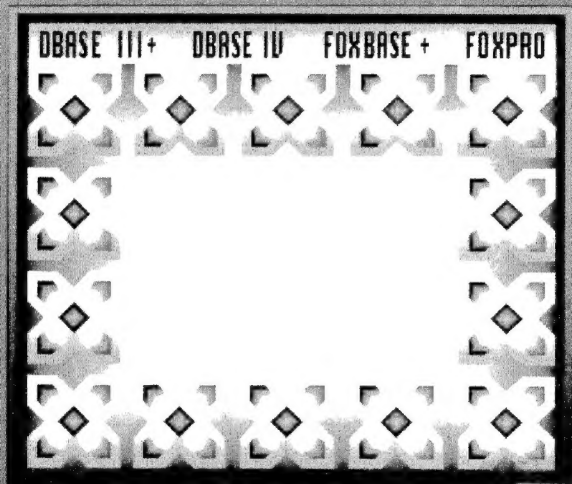


# نظم إدارة قواعد البيانات

الجزء الثاني



أ.د. محمد فهمي طلبة

أ.د. عمرو جتيد

أ.د. محمد علي الشرقاوي

مصطفى رضا عبد الوهاب

علاء الدين محمد فهمي

مصطفى محمد إسماعيل

٦

موسوعة دلتا كمبيوتر



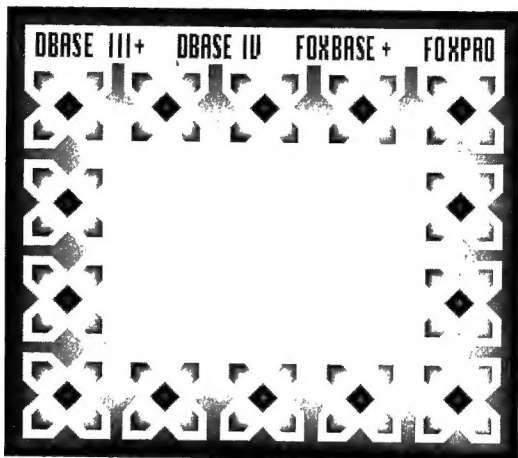


**نظم إدارة قواعد البيانات**



# نظم إدارة قواعد البيانات

الجزء الثاني



م . مصطفى رضا عبد الوهاب	ا . د . محمد فهمي طلبه
د . علاء الدين محمد فهمي	ا . د . عمرو جنييد
مصطفى محمد إسماعيل	ا . د . محمد علي الشرقاوي

٦

موسوعة دلتا كمبيوتر

### © حقوق النشر

لا يجوز نشر أى جزء من هذا الكتاب أو اختزان مادته بطريقة الاسترجاع ، أو نقله على أى وجه ، أو بأى طريقة ، سواء كانت إلكترونية ، أو ميكانيكية ، أو بالتصوير ، أو بالتسجيل ، أو خلاف ذلك إلا بموافقة الناشر على هذا كتابة ومقدمًا .

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the publisher.

رقم الإيداع ١٩٩١/٢٦٥٣

## تقديم

ان التطور في لغات البرمجة للحاسب الالكتروني قد مر بمراحل متعددة ارتبطت بعدة عوامل من أهمها التطور التكنولوجي المتلاحق في مكونات الحاسب المادية وما تبع ذلك من زيادة سرعة عمليات الحاسب المختلفة وزيادة كفاءتها . وقد أدى هذا الى تطور هائل في نظم التشغيل وظهور مفاهيم جديدة مثل تعدد الوظائف وتعدد المستخدمين واستخدام الشبكات بالإضافة الى العديد من الخصائص التي تميز الاجيال المتلاحقة من الحاسب الالكتروني .

ومن الملامح الرئيسية لتطور لغات البرمجة البعد التدريجي عن التدخل في العديد من العمليات التفصيلية التي تتم بواسطة الحاسب لاداء مهمة معينة مما أتاح الفرصة أمام مخططي البرامج لاستخدام خصائص الحاسب دون الحاجة الى الالام الكامل بعملياته الداخلية . ومن المعروف أن الحاسب حقيقة لا يفهم الا لغة الواحد والصفر وهي ما يطلق عليها لغة الماكينة (Machine Language) . وكان مخطط البرامج لا يستطيع التفاهم مع الحاسب الا من خلال هذه اللغة المعقدة . ثم ظهرت لغات أكثر سهولة قامت بتكوين مجموعات من الواحد والصفر في رموز بسيطة واستخدام هذه الرموز في كتابة البرامج . وتسمى هذه اللغات باللغات الرمزية (Symbolic Languages) او لغات التجميع (Assembly Languages) . وأخذت اللغات في التطور مع زيادة درجة التمثيل حتى وصلت الآن الى لغات الجيل الرابع (4th Generation Languages) حيث أتاحت هذه اللغات لمخطط البرامج التعامل مع النوافذ والقوائم الواضحة التي يستطيع من خلالها تصميم البرامج المطلوبة بسهولة تامة ويسر .

وقد انعكس هذا التطور الكبير على المتخصصين في مجال الحاسب حيث أصبح على مخططي البرامج متابعة كل جديد في مجال نظم تطوير البرامج وأدواتها المتقدمة وذلك حتى يمكنهم الاستفادة من خصائصها في تصميم النظم المتميزة التي توفر الكفاءة العالية وسهولة الاستخدام . ومن ناحية أخرى فقد أصبح على المستخدم ضرورة الالام بهذه البرامج التطبيقية الحديثة حتى يستطيع الاعتماد على نفسه في الاستفادة منها بالإضافة اليها . وقد أدى ذلك الى ظهور جيل جديد من المستخدمين الذين يمتلكون خبرة كبيرة في التعامل مع العديد من البرامج التطبيقية الى جانب القدرة على تصميم النظم الخاصة بهم .

وقد كان لهذا التطور في مجال الحاسبات اثره في طبيعة الكتب المتخصصة التي ظهرت على المستوى العالمي في الآونة الاخيرة . فبعد أن كانت هذه الكتب - الى وقت قليل مضى - تركز على الجوانب النظرية ، أصبحت الآن تركز على أساليب استخدام التطبيقات وعلى تقديم الخبرات والمهارات العملية لمستخدمي الحاسبات .

وقيام مؤسسة "دلتا" بتقديم موسوعتها الجديدة في تكنولوجيا وعلوم الحاسب يعوض النقص الشديد الذي تعاني منه المكتبة العربية في هذا المجال حيث أن معظم الكتب العربية الموجودة ليست سوى ترجمة أو تلخيص سطحي لدليل التشغيل للنظم المختلفة بينما يحتاج المستخدم الى توضيح الكثير من الجوانب العلمية والفنية

بالإضافة إلى خصائص تشغيل النظم . وهذا الجهد من مؤسسة دلتا هو امتداد لسياستها الواعية واحساسها بمسئوليتها نحو التطور التكنولوجي بالمنطقة العربية .

وهذا الكتاب هو أحد كتب موسوعة "دلتا" لتكنولوجيا و علوم الحاسب والذي يمثل حلقة الاتصال بين الجوانب التطبيقية والجوانب العلمية والفنية . فبالرغم من تركيزه على شرح خصائص تشغيل نظم عائلة (DBase) ، إلا أنه اهتم بتوضيح مفهوم قواعد البيانات بصفة عامة وتوضيح العمليات المرتبطة بها مثل الفهرسة والبحث عن البيانات وأساسيات تصميم البرامج ، و ..... الخ .

ومما لاشك فيه أن موضوع هذا الكتاب يعد من أهم الموضوعات التي تشغل أذهان جميع المتخصصين في مجال الحاسب وذلك لارتباطه المباشر بالمجالات العملية سواء للأفراد أو للمؤسسات . فقد أصبحت قواعد البيانات وتطبيقاتها تغطي معظم مجالات الحياة وأصبح استخدام الحاسب في إدارة البيانات والسيطرة عليها أمراً مألوفاً في كل موقع . ورغم توفر عدد كبير من البرامج التطبيقية التي تخصصت في إدارة قواعد البيانات ، مثل (Informex) ، (Oracle) ، (Paradox) ، (Focus) ، ..... الخ إلا أن مؤسسة "دلتا" اختارت برنامج عائلة (DBase) لتكون موضوع هذا الكتاب بأجزائه الثلاث وذلك لما تتمتع به عائلة (DBase) من الشبوع والانتشار وسهولة الاستخدام بالإضافة إلى قدرتها على التعامل مع العديد من نظم التشغيل .

١. د. محمد فهمي طلبه



## محتويات الكتاب

مستند	الموضوع	رقم الصفحة
١	مقدمة	١
٢	الباب السابع والعشرون " أهم الأوامر المستخدمة "	٣
٣	الباب الثامن والعشرون " أهم الدوال المستخدمة "	١٣١
٤	ملحق (١) " أهم الأوامر والدوال المستخدمة في برنامج (DBase IV) "	١٧٧
٥	أولا : الأوامر	١٧٩
٦	ثانيا : أوامر التجهيز	١٩٠
٧	ثالثا : الدوال	١٩٥
٨	ملحق (٢) " أهم الأوامر والدوال المستخدمة في برنامج (FoxBase +) "	٢٠١
٩	أولا : الأوامر	٢٠٣
١٠	ثانيا : الدوال	٢٠٥
١١	ملحق (٣) " أهم الأوامر والدوال المستخدمة في برنامج (FoxPro) "	٢١١
١٢	أولا : الأوامر	٢١٣
١٣	ثانيا : الدوال	٢١٥



## مقدمة

هذا الكتاب هو الجزء المكمل للكتاب الأول الذي نشر تحت اسم " نظم ادارة قواعد البيانات " - الجزء الأول . ورغم أن الكتاب الأول يتضمن الشرح التفصيلي لخصائص برامج عائلة (DBase) بالإضافة الى أساسيات كتابة البرامج بواسطتها وهذا الشرح يعتبر كافيا للامام الكامل بالخصائص الفنية لهذه البرامج الا أن مؤسسة "دلتا" رأت استكمالا لهذا الجزء أن توفر للمستخدم مرجعا شاملا لجميع الأوامر والدوال المستخدمة في عائلة (DBase) . متضمننا الأوامر والدوال الخاصة ببرامج (DBase III +) ، (DBase IV +) ، (FoxBase +) ، (FoxPro) .

وهذا المرجع يعتبر ضروريا لمن يريد اكتساب خبرة كبيرة ببرامج عائلة (DBase) والاعتماد على نفسه في تصميم النظم الكاملة باستخدام برامج العائلة . حيث أن المستخدم أثناء اعداده للبرامج يحتاج دائما الى الرجوع الى أمر معين ومعرفة الشكل (syntax) الخاص به بالإضافة الى خصائص استخدام هذا الأمر . وهذا المرجع الشامل يتيح له الوصول مباشرة الى الأمر المطلوب . حيث أن الأوامر مرتبة حسب الترتيب الهجائي للحروف . وسوف يجد القارئ الشرح الوافي لكل أمر متضمننا الرسم التوضيحي الملائم والأمثلة المناسبة .

ونظرا لأن هذا الكتاب هو جزء مكمل للكتاب الأول كما سبق الايضاح ، فقد تم ترتيب الأبواب بتسلسل مكمل للتسلسل الخاص بالكتاب السابق . لذلك يبدأ الكتاب بالبواب رقم (٢٢) . مع ملاحظة أن الأوامر والدوال مرتبة حسب الترتيب الهجائي للحروف الانجليزية .



## **الباب السابع والعشرون**

### **أهم الأوامر المستخدمة**



ان قائمة الأوامر المستخدمة في كتابة البرامج عن طريق برنامج (Dbase III+) وكذلك باقي برامج عائلة (Dbase) مثل (Dbase IV) ، (FoxBase +) ، (FoxPro) كبيرة جدا ولن يتسع المجال لدراستها بالتفصيل في هذا الكتاب . ولكن سيتم في هذا الباب القاء الضوء على معظم هذه الأوامر مع دراسة تفصيلية لها . كلما أمكن . مع ملاحظة أن الأوامر مرتبة حسب الترتيب الهجائي للحروف الانجليزية . كما يستطيع القارئ الرجوع الى الملحق الموجود في نهاية هذا الكتاب لمعرفة الأوامر والدوال الاضافية لبرامج (Dbase IV) ، (FoxBase +) ، (FoxPro) .

## ملاحظة

القيم الموجودة داخل قوسين مربعين [ ] هي قيم اختيارية يستطيع المستخدم كتابتها او عدم كتابتها . كذلك فان مايكتب بين الزاويتين < > يعنى عناصر مختارة بواسطة المستخدم نفسه . اما الزاويتين نفسهما فلا يكتبان ضمن العنصر .

## ١ - الأمر (?)

يستخدم هذا الأمر لعرض محتويات متغير ذاكرة أو حقل معين في ملف قاعدة البيانات عن طريق كتابة اسم هذا المتغير أو الحقل بعد علامة الاستفهام . وهو يعنى استفهام أو سؤال عن قيمة مطلوبة . وتتم الكتابة من أول السطر التالى .

والصورة العامة للأمر كالاتى :

[ < expression list > ] ?

حيث (expression List) هي اسم متغير ذاكرة أو حقل أو أى علاقة مطلوب حساب قيمتها .

وعند كتابة هذا الأمر بدون أى علاقة بعده فان هذا يؤدي الى عرض سطر خال . وتستخد هذه العملية عندما يراد عرض سطور خالية بين المخرجات المطبوعة .

## أمثلة

لحساب نتيجة معادلة حسابية يتم كتابة السطر التالى :

? 2 \* 2 + (8/2)

@

ولعرض تاريخ اليوم الحال يتم كتابة السطر التالي :

. ? DATE()

01/07/90

ولعرض بيانات حقول في قاعدة البيانات يتم كتابة السطور التالية :

. USE Cadets

. ? name , address

Mohamed Aly 12 - Nasr City - Cairo

## ٢ - الأمر (??)

هذا الأمر يؤدي نفس العمل مثل الأمر السابق تماما ، ولكنه يسمح بالكتابة من أى مكان في السطر ولا يشترط الكتابة من أول السطر .

## ٣ - الأمر @

هذا الأمر يستخدم في انشاء شاشات الادخال والتقارير بالاضافة الى التحكم في مكان عرض البيانات على الشاشة .

وهناك عدة صور لهذا الأمر يتم دراسة كل منها على حدة . وهي :

### ١ - الصورة الأولى :

@ <row, col> SAY <expression list>

وتستخدم هذه الصورة في عرض أى بيانات موجودة في متغيرات ذاكرة ، أو حقول داخل ملف قاعدة البيانات ، أو رسائل خاصة يراد عرضها علي المستخدم . ويتم ذلك عن طريق كتابة الاحداثيات المطلوب عرض البيانات بدءا منها . وهذه الاحداثيات هي رقم السطر (Row) ورقم العمود (Column) .



## مثال

```
USE Cadets
@ 10,10 SAY name
```

وعند تنفيذ هذه الأوامر يظهر السطر التالي مثلا :

Ahmed Salem

وذلك بدءا من السطر العاشر والعمود العاشر .

## ملاحظة

الرقم المقابل للسطر (row) يأخذ أى قيمة من صفر الى ٢٢ ، والرقم المقابل للعمود (col) يأخذ أى قيمة من صفر الى ٧٩ . مع ملاحظة أن ترقيم السطور يبدأ من أعلى ، وترقيم الأعمدة يبدأ من اليسار . ويمكن تحويل السطور المكتوبة بواسطة هذا الأمر الى الطباعة عن طريق كتابة الأمر (SET DEVICE TO PRINT) .

## ب - الصورة الثانية

```
@ <row,col> SAY <expression list> PICTURE <clause>
```

وتستخدم هذه الصورة كالصورة السابقة تماما مع اضافة التحكم في شكل البيانات المعروضة عن طريق الأمر (PICTURE) . وبلى هذه الكلمة تعبير (Clause) يحدد صورة هذه البيانات . ويتم استخدام نموذج (Template) مكان هذا التعبير (Clause) . وهذا النموذج قد يكون مجموعة من الرموز (Symbols) يمثل كل منها حرفا من حروف البيانات التي تظهر في هذا السطر . وقد تكون دالة معينة (Function) يتم ادخالها للتحكم في شكل جميع الحروف مرة واحدة .

## رموز النموذج (Template Symbols)

وهي رموز يستخدم بعضها مع الأمر (@...SAY) ، ولكنها تستخدم في الغالب مع الأمر (@...GET) كما سيتم الايضاح . والجدول التالي يوضح هذه الرموز ومعنى كل منها :

- 9 ويسمح بظهور الأعداد فقط .  
 \* ويسمح بظهور الأعداد وعلامات الجمع والطرح والمسافات بين الأرقام .  
 A ويسمح فقط بظهور الحروف الهجائية .  
 L ويسمح بظهور البيانات المنطقية .  
 Y ويسمح بظهور البيانات المنطقية Y , y , N , n .  
 N ويسمح بكتابة الحروف أو الأرقام .  
 X ويسمح بكتابة أى حروف أو حروف خاصة .  
 ! وهو يحول الحروف الصغيرة الى كبيرة (Capital) .  
 \$ وهو يعرض علامة الدولار قبل العدد .  
 . وهو يحدد مكان العلامة العشرية (Decimal Point) .

### دوال النموذج (Template Functions)

كما سبق الايضاح فان هذه الدوال تتحكم في شكل الحروف بالكامل . ولايلزم كتابة دالة لكل حرف . ويمكن الجمع بين الدالة (Function) والرموز (Symbols) في نفس التعبير بعد الأمر (PICTURE) على أن يتم كتابة الدالة أولاً ثم الرموز . وتبدأ الدالة عادة بالحرف @ . كما يمكن كتابة كلمة (FUNCTION) بدلاً من الحرف @ حتى لا يحدث خلط بينه وبين الأمر @ الموجود في أول السطر .

والجدول التالي يوضح كل دالة ومعنى كل منها :

- C وهي تعرض الحروف (CR) أى (Credit) بعد الأعداد السالبة .  
 ( وهي تعرض الأعداد السالبة بين قوسين .  
 B وهي تؤدي الى ضبط الأعداد من اليسار .  
 Z وهي تؤدي الى حذف الأرقام التي قيمتها صفر .  
 D وهي تؤدي الى عرض التواريخ بالصورة الأمريكية .  
 E وهي تؤدي الى عرض التواريخ بالصورة الأوروبية .  
 A وهي تؤدي الى عرض الحروف الهجائية فقط .  
 ! وهي تؤدي الى تحويل الحروف الى حروف كبيرة (Capital) .  
 R وهي تؤدي الى عرض حروف خاصة بين البيانات المعروضة .  
 <n>S وهي تؤدي الى تحديد عرض البيانات المعروضة بعدد (n) من الحروف . وتسمح بهزحزة الحروف (Scrolling) خلال هذا العرض .

## أمثلة

السطر التالي يؤدي إلى ضبط العدد الموجود في الحقل (amount) جهة اليسار (Left Justified) وذلك عن طريق الدالة (B) .

```
@ 5,5 SAY amount PICTURE '@B 9,999,999.99'
```

كما يمكن كتابة نفس السطر السابق بصورة أخرى كالآتي :

```
@ 5,5 SAY amount FUNCTION 'B';  
PICTURE '9,999,999.99'
```

ويمكن كتابة عدة دوال مع نفس الصورة (PICTURE) كالآتي :

```
@ 5,5 SAY amount PICTURE '@ XC 999.99'
```

## ج. الصورة الثالثة

```
@<row,col> GET <variable> PICTURE <clause>  
RANGE <exp>,<exp>
```

وتستخدم هذه الصورة في عرض عمود ضوئي على الشاشة يمثل المتغير (variable) المكتوب . ومن خلال هذا العمود الضوئي يستطيع المستخدم ادخال البيانات المطلوبة والتي يتم تخزينها في المتغير . ويجب ملاحظة أن هذا المتغير يلزم انشاؤه أولا قبل كتابة هذا الأمر .

وكلمة (PICTURE) تم شرحها في الصورة السابقة ، وهي تؤدي هنا إلى التحكم في شكل البيانات التي يدخلها المستخدم . كما تؤدي إلى تحويلها إلى الشكل المطلوب ادخاله في المتغير .

وكلمة (RANGE) تستخدم مع المدخلات العددية والتاريخية لتحديد أقل قيمة وأكبر قيمة مطلوب ادخالها في المتغير .

ولتحديد مدى تاريخي مثلا يجب أولا تحويل التاريخ من الحروف إلى تاريخ عن طريق الدالة (CTOD) . فمثلا يمكن كتابة المدى بين تاريخين كالآتي :

RANGE CTOD('01/01/90'), CTOD('02/05/90')

وهذا يحدد المدى من ١٠/١/١ الى ١٠/٢/٥

ويمكن كتابة الحد الأدنى فقط أو الحد الأقصى فقط كالآتي مثلا:  
(Range 30,) . وهذا يعني أى عدد يزيد من ( ٢٠ ) . أما  
(Range , 100) فهو يعني أى عدد أقل من ( ١٠٠ ) .

#### د - الصورة الرابعة

@ <row1, col1> TO <row2, col2> [ DOUBLE ]

وتستخدم هذه الصورة في رسم مستطيل يبدأ من النقطة الممثلة  
بالاحداثيات <row1, col1> التي تمثل أعلى نقطة يسار هذا المستطيل  
وينتهي بالنقطة الممثلة بالاحداثيات < row2, col2 > التي تمثل أدنى نقطة  
يمين المستطيل .

واستخدام (DOUBLE) اختياري وهو يؤدي الى رسم المستطيل بخطوط  
مزدوجة .

وتفيد هذه الصورة في رسم اشكال هندسية في شاشة ادخال البيانات  
تعطي الشاشة شكلا مثيرا وجذابا .

#### أمثلة

- السطر التالي يؤدي الى رسم مستطيل بخطوط مفردة (Single) .

@ 2,20 TO 8,60

- السطر التالي يؤدي الى رسم مستطيل بخطوط مزدوجة (Double) .

@ 1,10 TO 7,50 DOUBLE

- السطر التالي يؤدي الى رسم خط أفقي

@ 3,5 TO 3,30

- وذلك لأن رقم السطر ثابت في النقطتين .
- السطر التالي يؤدي الى رسم خط رأسي

@ 1,20 TO 20,20

- وذلك لأن رقم العمود ثابت في النقطتين .

### هـ - الصورة الخامسة

@ <row1, col1> CLEAR TO <row2, col2>

وتستخدم هذه الصورة في مسح مستطيل من الشاشة يبدأ من النقطة < row1, col1 > وينتهي بالنقطة < row2, col2 > . ويمكن استخدام هذه الصورة في مسح أجزاء مختلفة من الشاشة حسب الحاجة . كما يمكن مسح سطر واحد عن طريق كتابة أول نقطة في هذا السطر وآخر نقطة في السطر .

### مثال

@ 1,0 CLEAR TO 6,50

وهذا الأمر يؤدي الى مسح مستطيل يبدأ من النقطة ( 0 , 1 ) وينتهي بالنقطة ( 50 , 6 ) .

### ٤ - الأمر (ACCEPT)

يستخدم هذا الأمر في عرض رسالة للمستخدم واستقبال قيمة معينة يدخلها المستخدم رداً على هذه الرسالة . حيث يتم تخزين هذه القيمة في متغير ذاكرة يتم انشاؤه من خلال هذا الأمر .

والصورة العامة لهذا الأمر كالآتي :

ACCEPT [<message>] TO <memvar>

ويجب ملاحظة أن الرسالة (message) في هذه الحالة يمكن أن تكون متغير ذاكرة حرفي (Character) أو تكون سلسلة حرفية يتم كتابتها بين علامات تنصيص (Quotation).

كما يجب ملاحظة أن المتغير (memvar) يكون دائما متغيرا حرفيا (Character).

## مثال

```
ACCEPT "Enter your name:" TO mname
```

في هذه الحالة تظهر الرسالة التالية للمستخدم :

```
Enter your name :
```

وعندما يكتب المستخدم اسمه ويضغط على مفتاح الادخال يتم تخزين هذا الاسم في متغير الذاكرة (mname).

## ٥ - الأمر (APPEND)

يستخدم هذا الأمر في اضافة سجلات جديدة الى ملف قاعدة البيانات . وهو يستخدم شاشة الادخال المستخدمة سواء كانت شاشة الادخال المبدئية (Default) أو شاشة الادخال التي يتم تصميمها من خلال البرنامج .

والصورة العامة للأمر كالآتي :

```
APPEND [ BLANK ]
```

ويستخدم الاختيار (BLANK) لاضافة سجل خال في نهاية ملف قاعدة البيانات حتى يمكن اخلال بيانات الحقول مكان الحقول الخالية من خلال البرنامج .

## مثال

يمكن كتابة الأوامر التالية :

.USE Cadets

.APPEND

وعند الضغط على مفتاح الادخال تظهر الشاشة كما في المثال بشكل ( ٢٧ - ١ )

NAME	<input type="text"/>	TELEPHONE	<input type="text"/>
ADDRESS	<input type="text"/>	CLASS	<input type="text"/>
NATIONALITY	<input type="text"/>	HOBBIES	<input type="text"/>
FATHER NAME	<input type="text"/>	BIRTH DATE	<input type="text"/>
FATHER JOB	<input type="text"/>	RELIGION	<input type="text"/>
MOTHER NAME	<input type="text"/>	NOTES	<input type="text"/>

شكل ( ٢٧ - ١ )

وعند الانتهاء من ادخال بيانات الحقول يتم الضغط على مفتاح الادخال فيتم تخزين هذا السجل . كما يمكن الضغط على مفتاحي (Ctrl-End) لتخزين جميع السجلات التي تمت اضافتها والعودة الى مشيرة النقطة (Dot Prompt) .

## ٦ - الأمر (APPEND FROM)

يستخدم هذا الأمر في نسخ سجلات من ملف الى نهاية ملف قاعدة البيانات المفتوح . ولا يشترط أن يكون الملف المنسوخ منه ملف قاعدة بيانات كما سنلاحظ من الصور المختلفة للأمر . وانما يمكن أن يكون ملف جدول الكتروني (Spread Sheet) أو أي ملف آخر مكتوب بشفرة الآسكي (ASCII Code) .

وهناك صورتان للأمر ، الصورة الأولى تستخدم للنسخ من ملف قاعدة بيانات آخر، والصورة الثانية تستخدم للنسخ من ملفات أخرى .

## 1 - الصورة الأولى

APPEND FROM <Filename> [FOR <condition>]

حيث (filename) هو اسم الملف المنسوخ منه . ولا يكتب فيه الامتداد (Extension) حيث أن البرنامج يضيف الامتداد (.dbf) ألياً .

و (condition) هو الشرط الذي يحدد السجلات المطلوب نقلها ويجب ملاحظة أن الحقول المشتركة في الملفين فقط هي التي يتم نسخ بياناتها . ولا يشترط أن تكون بنفس الترتيب في الملفين . وإذا كان أى حقل في الملف المنسوخ منه أكبر من الحقل المقابل في الملف الآخر ، يتم حذف الحروف الزائدة (Truncation) إذا كان الحقل حرفياً ، كما يتم استبدال الأرقام بنجوم (Astisks) إذا كان الحقل عددياً .

### مثال

نفرض أنه قد تم إنشاء ملف قاعدة بيانات اسمه (School1) ويراد إضافة بيانات الطلبة الناجحين فقط في المدرسة الى ملف آخر اسمه (Cadets) . لتنفيذ ذلك يتم كتابة الأوامر التالية :

USE Cadets

APPEND FROM School1 FOR grade>50

في هذه الحالة يتم إضافة السجلات التي تحقق الشرط فقط الى الملف (Cadets) .

## ب - الصورة الثانية

APPEND FROM <Filename> TYPE <Filetype>

وتستخدم هذه الصورة في إضافة سجلات من ملفات ليست مكتوبة بواسطة (DBase III+) وهي تشمل الأنواع التالية :

١ - الملفات ذات الامتداد (SDF) وهي الملفات المكتوبة بشفرة الآسكي (ASCII Code) . ويتم نسخها حرفاً حرفاً بحيث ينتهي كل سجل بالكود الخاص بالادخال (Carriage Return) .



٢ - الملفات ذات الامتداد (SYLK) وهي الملفات المكتوبة بواسطة برنامج (VisiCalc) ، وهو برنامج جداول الكترونية . حيث يتم تحويل السطور (rows) الى سجلات والأعمدة (Columns) الى حقول .

٢ - الملفات ذات الامتداد (WKS) وهي الملفات المكتوبة بواسطة برنامج لوتس ١٢٢ (Lotus 123) حيث يتم تحويل السطور الى سجلات والأعمدة الى حقول .

ويجب ملاحظة كتابة الامتداد عند كتابة اسم الملف المنسوخ منه (Filename) ، وكذلك كتابة نوع الملف المنسوخ منه بعد كلمة (TYPE) وذلك كالآتي مثلا :

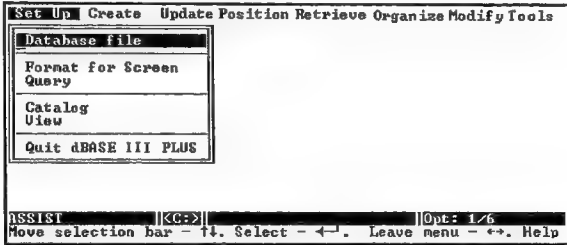
APPEND FROM ACCOUNTS.WKS TYPE WKS

## ٧ - الأمر (ASSIST)

يستخدم هذا الأمر في تشغيل القوائم الرئيسية لبرنامج المساعد (ASSISTANT) التي يمكن من طريقها انشاء ملف قاعدة البيانات والملفات المرتبطة به مثل ملفات الفهرس (Index Files) ، وملفات التشكيل (Format Files) ، وملفات البحث (Query Files) ، وملفات النظر (View Files) ، بالإضافة الى ملفات التقارير (Reports) والعناوين المختصرة (Labels) . وتسمح هذه القوائم كذلك بتعديل هذه الملفات وأجراء عمليات التصحيح والبحث والعرض لأي بيانات مطلوبة .

ويتم تنفيذ أي اختيار من القوائم عن طريق تحريك مؤشر القائمة حتي يصل الى الاختيار المطلوب ثم الضغط على مفتاح الإدخال . ويلاحظ عند الوصول الى الاختيار النهائي المطلوب تنفيذه ، ظهور الأمر المقابل له بالكامل على السطر أعلى عمود الحالة (Status Bar) والذي يسمى سطر الفعل (Action Line) . وهذا الأمر الذي يظهر هو نفس الأمر الذي يمكن للمستخدم كتابته من مشيرة النقطة (Dot Prompt) للحصول على نفس النتيجة . وتتيح هذه الطريقة للمستخدم التعرف على الشكل (Syntax) الخاص بأي أمر أثناء العمل من خلال قوائم برنامج المساعد (Assistant) . كما يمكن الحصول على أي معلومات اضافية على الشاشة . وذلك عن طريق عرض شاشة مساعدة (Help) توضح للمستخدم شكل الأمر (Syntax) بالإضافة الى شرح خصائص استخدام هذا الأمر .

ويمكن تشغيل قوائم برنامج المساعد (Assistant) عن طريق كتابة الأمر (ASSIST) من مشيرة النقطة (Dot Prompt). كما يمكن تشغيلها أيضا عن طريق الضغط على مفتاح (F2). ويلاحظ في هذه الحالة ظهور القائمة الرئيسية على الشاشة. انظر الشكل ( ٢٧ - ٢ ).



شكل ( ٢٧ - ٢ )

وهذه القائمة تحتوي على ثمانية اختيارات يمكن تلخيصها كالآتي :

#### ١ - قائمة التجهيز (Set Up)

تستخدم هذه القائمة في فتح ملف قاعدة البيانات والملفات المرتبطة به . كما تستخدم أيضا في الخروج من البرنامج . وعند فتح أى ملف تظهر قائمة بوحدة الأقراص المتاحة لاختيار وحدة الأقراص التي تحتوي على الملف المطلوب فتحه .

وعند فتح ملف قاعدة البيانات وبعد اختيار وحدة الأقراص والملف المطلوب ، يظهر سؤال عما إذا كان الملف مفهرسا (Indexed) أم لا . فإذا كان الملف مفهرسا يتم كتابة (Y) فتظهر قائمة بأسماء ملفات الفهرس ويتم اختيار ملفات الفهرس الخاصة بملف قاعدة البيانات المفتوح على ألا يزيد عدد ملفات الفهرس المفتوحة عن سبعة ملفات . وكل ملف يتم اختياره تظهر أمامه

علامة ( ► ) ، مع ملاحظة أن أول ملف يتم اختياره يصبح هو ملف الفهرس الرئيسي (Master) بصرف النظر عن ترتيب هذا الملف في القائمة . ويتم تخزين هذه الاختيارات عن طريق الضغط على مفتاح السهم شمال (←) .

وعند فتح ملف التشكيل (Format) يتم تحديد وحدة الأقراص فتظهر قائمة بملفات التشكيل الموجودة . ويتم اختيار ملف التشكيل المطلوب استخدامه .

ويمكن فتح ملف البحث (Query) وذلك باختيار ملف البحث المطلوب من قائمة الملفات التي تظهر على الشاشة .

كما يمكن فتح الكتالوج اذا كان قد سبق انشاء ملف كتالوج حيث يتم اختيار ملف الكتالوج المطلوب من قائمة الملفات التي تظهر على الشاشة . وفي هذه الحالة لا تظهر على الشاشة دائما الا الملفات المخزنة في هذا الملف . فعند فتح ملف قاعدة البيانات أو الملفات المرتبطة به ، لا تظهر الا الملفات الموجودة في ملف الكتالوج المفتوح . في حين لو لم يتم فتح ملف الكتالوج تظهر جميع الملفات الموجودة على وحدة الأقراص المستخدمة (Current Drive) .

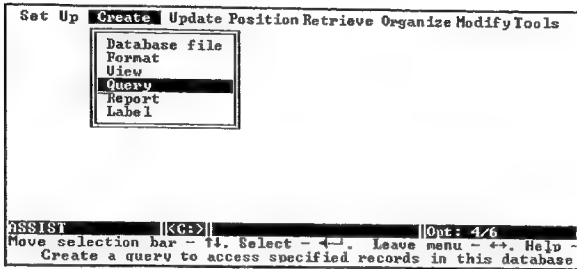
ويمكن فتح ملف المنظر (View File) بنفس الطريقة مثل الملفات السابقة . حيث يتم تحديد وحدة الأقراص المستخدمة ثم اختيار ملف المنظر من قائمة ملفات المنظر التي تظهر على الشاشة .

وللخروج من البرنامج يتم اختيار (Quit DBase III+) ويؤدي هذا الى اغلاق جميع الملفات المفتوحة والعودة الى نظام التشغيل .

## ب - قائمة الانشاء (Create)

تستخدم هذه القائمة في انشاء الملفات . ولكل نوع من الملفات يتم اختيار وحدة الأقراص المطلوب تخزين الملف فيها ، ثم يتم كتابة اسم الملف . وإذا كان قد سبق فتح كتالوج ، فان هذا الملف الذي يتم انشاؤه يدخل في الكتالوج .

انظر شكل ( ٢٧ - ٢ )



شكل ( ٢٧ - ٢ )

والاختيار (Database File) يستخدم في إنشاء ملف قاعدة بيانات .

ارجع الى الأمر (CREATE) .

والاختيار (Format) يستخدم في إنشاء ملف تشكيل (Format File) . وهو ملف يؤدي الى التحكم في شكل شاشة الادخال التي يتم عن طريقها ادخال البيانات .

ارجع الى الأمر (CREATE SCREEN) .

والاختيار (View) يستخدم في إنشاء ملف المنظر (View File) وهو يسمح بالتعامل مع عدة ملفات قواعد بيانات في نفس الوقت .

ارجع الى الأمر (CREATE VIEW)

والاختيار (Query) يستخدم في إنشاء ملف بحث (Query File) وهذا يساعد على ترشيح ملف قاعدة البيانات (Filtering) للحصول على البيانات المطلوبة .

ارجع الى الأمر (CREATE QUERY) .

والاختيار (Report) يستخدم فى انشاء ملف التقرير الذى يحتوى على الامتداد (.Frm) .

ارجع الى الامر (CREATE REPORT) .

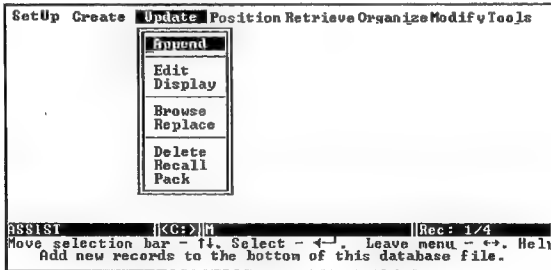
والاختيار (Label) يستخدم فى انشاء ملف العناوين المختصرة (Label) .

ارجع الى الامر (CREATE LABEL) . .

### ج - قائمة التحديث (Update)

وتستخدم هذه القائمة فى تحديث البيانات المخزنة فى قاعدة البيانات . فيمكن من خلالها اضافة سجلات أو مسح سجلات أو تصحيح بيانات سجلات معينة فى الملف . وعند فتح ملفات الفهرس (Index Files) فان هذه الملفات يتم تحديثها كليا مع تحديث بيانات قاعدة البيانات .

انظر شكل ( ٢٧ - ٤ )



شكل ( ٢٧ - ٤ )

والاختيار (Append) يستخدم فى اضافة سجلات جديدة فى نهاية الملف .

• ارجع الى الأمر (APPEND) .

والاختيار (Edit) يستخدم في تعديل بيانات الملف سجلا سجلا ، وذلك حسب السجل الذى يقف عنده مؤشر الملف .

• ارجع الى الأمر (EDIT) .

والاختيار (Display) يستخدم في عرض بيانات سجل معين .

• ارجع الى الأمر (DISPLAY) .

والاختيار (Browse) يستخدم في تعديل بيانات الملف ، بالاضافة الى عرض حتى ١٧ سجلا على الشاشة .

• ارجع الى الأمر (BROWSE) .

والاختيار (Replace) يسمح باجراء تعديلات مجمعة (Batch) لملف قاعدة البيانات . ويمكن استخدامه مثلا في تعديل مرتب جميع الموظفين في شركة معينة عن طريق ضرب حقل المرتب (Salary) في نسبة ثابتة مثل (5 %).

والاختيار (Delete) يستخدم في وضع علامات (Marks) على السجلات المطلوب مسحها تمهيدا لمسحها تماما بواسطة الاختيار (Pack) .

والاختيار (Recall) يستخدم في استعادة السجلات التي تم وضع علامات عليها لمسحها . ويمكن في هذه الحالة استعادة بعض السجلات أو كلها حسب الحاجة .

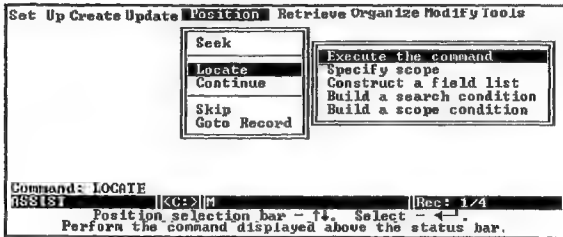
والاختيار (Pack) يستخدم في المسح النهائي للسجلات التي تم وضع علامات عليها لمسحها .

د - قائمة المكان (Position)

وتستخدم هذه القائمة في توجيه مؤشر السجلات (Record Pointer) الى سجل محدد ، وذلك حتي يمكن تحديث هذا السجل أو عرض بياناته . وفي كل اختيار من اختيارات هذه القائمة يقوم البرنامج بعرض القوائم الفرعية التي

يتم عن طريقها ادخال الشرط أو الشروط التي يتم بناء عليها توجيه المؤشر الى سجل محدد .

انظر شكل ( ٢٧ - ٥ )



شكل ( ٢٧ - ٥ )

والاختيار (Seek) يستخدم فقط عندما يكون الملف مفهرسا (Indexed) . وهو يوجه مؤشر السجلات الى أول سجل يحقق الشرط الذي يتم ادخاله .

ارجع الى الأمر (SEEK) .

والاختيار (Locate) يستخدم في توجيه مؤشر السجلات (Record Pointer) الى أول سجل يحقق الشرط الذي يتم ادخاله .

ارجع الى الأمر (LOCATE) .

والاختيار (Continue) يستخدم فقط بعد استخدام الاختيار (Locate) . وذلك لعرض السجل الثاني الذي يحقق الشرط الذي سبق ادخاله .

والاختيار (Skip) يستخدم في تحريك مؤشر السجلات (Record Pointer) إلى سجلات تالية أو سجلات سابقة حسب العدد الذي يتم ادخاله .

ارجع الى الأمر (SKIP) .

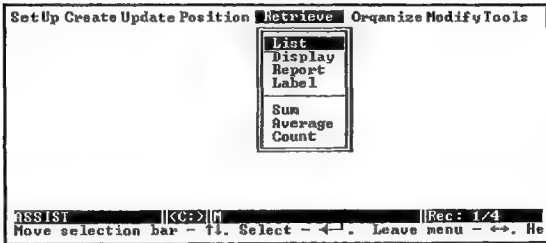
والاختيار (Goto Record) يستخدم في تحديد رقم سجل معين يراد وضع المؤشر عنده .

ارجع الى الأمر (GOTO) .

### هـ - قائمة الاسترجاع (Retrieve)

وتستخدم هذه القائمة في عرض بيانات سجلات معينة على الشاشة أو طباعتها على الطابعة .

انظر الشكل ( ٢٧ - ٦ )



شكل ( ٢٧ - ٦ )

والاختيار (List) يستخدم لعرض بيانات جميع السجلات المطلوبة على الشاشة أو طباعتها على الطابعة حسب الحاجة .

ارجع الى الأمر (LIST) .



والاختيار (Display) يستخدم لنفس الهدف مثل الاختيار (List) ولكن هناك بعض الاختلافات بينهما .

ارجع الى الأمر (DISPLAY) .

والاختيار (Report) يستخدم في عرض البيانات على الشاشة أو طباعتها على الطابعة حسب شكل التقرير (Report) الذي سبق انشاؤه .

ارجع الى الأمر (REPORT) .

والاختيار (Label) يستخدم في عرض البيانات على الشاشة أو طباعتها على الطابعة حسب شكل التقرير المختصر (Label) الذي سبق انشاؤه .

ارجع الى الأمر (LABEL) .

والاختيار (Sum) يستخدم لتجميع الحقول العددية لمجموعة من السجلات التي يتم اختيارها .

ارجع الى الأمر (SUM) .

والاختيار (Average) يستخدم لحساب القيم المتوسطة لكل الحقول العددية في مجموعة من السجلات التي يتم اختيارها .

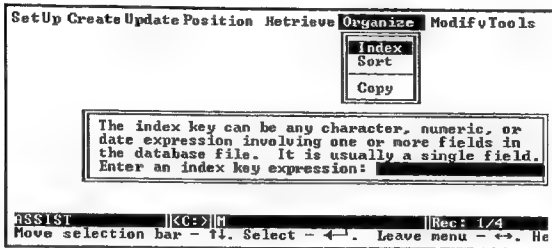
ارجع الى الأمر (AVERAGE) .

والاختيار (Count) يستخدم في حساب عدد السجلات التي تحقق شرطا أو شروطا معينة .

ارجع الى الأمر (COUNT) .

## و - قائمة التنظيم (organize)

تستخدم هذه القائمة في تنظيم السجلات وترتيبها داخل ملف قاعدة البيانات . وكل اختيار من اختيارات هذه القائمة يؤدي إلى انشاء ملف جديد .  
انظر الشكل ( ٢٧ - ٧ )



شكل ( ٢٧ - ٧ )

والاختيار (Index) يؤدي إلى إنشاء ملف فهرسي (Index File) . وعند فتح هذا الملف من خلال قائمة التجهيز (Setup) ، فإن سجلات ملف قاعدة البيانات تظهر دائما مرتبة حسب الحقل الفهرسي (Key Field) الذي تم اختياره وذلك رغم عدم حدوث أى تغيير فى الأماكن الفعلية للسجلات .

ارجع إلى الأمر (INDEX) .

والاختيار (Sort) يؤدي إلى إنشاء ملف قاعدة بيانات جديد مرتب بالترتيب المطلوب .

ارجع إلى الأمر (SORT) .

والاختيار (Copy) يستخدم فى عمل نسخة من ملف قاعدة البيانات . ويمكن نسخ الملف كله أو نسخ مجموعة من السجلات التى تحقق شرطا أو شروطا معينة فقط . كما يمكن أيضا نسخ بيانات مجموعة من الحقول فقط وليس كل الحقول .

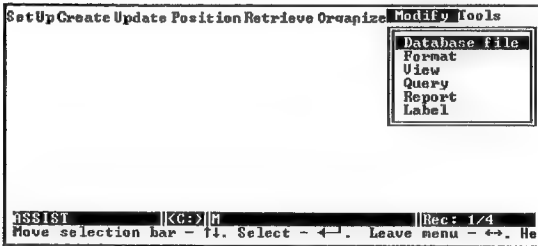
ارجع إلى الأمر (COPY) .

## ز - قائمة التعديل (Modify)

انظر الشكل ( ٢٧ - ٨ )

وهذه القائمة تستخدم في تعديل الملفات التي سبق انشاؤها من خلال قائمة الانشاء (Create) . وهي تحتوى على نفس الاختيارات الموجودة فى قائمة الانشاء (Create) ما عدا الكتلوج . وعند اختيار نوع الملف المطلوب تعديله سواء كان ملف قاعدة البيانات أو الملفات المرتبطة به ، فان البرنامج يعرض أسماء الملفات الموجودة على وحدة الأقراص من نفس النوع .

ارجع الى الأمر (MODIFY) .

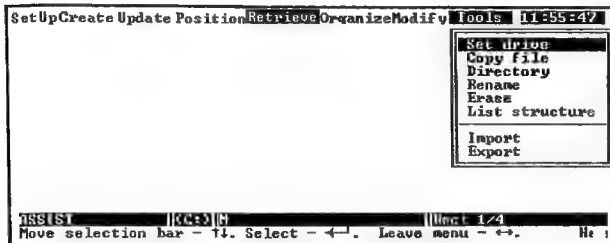


شكل ( ٢٧ - ٨ )

## ح - قائمة الأدوات (Tools)

وهذه القائمة تؤدي بعض الوظائف التي يقوم بها نظام التشغيل مثل عرض فهرس الملفات (Directory) ونسخ الملف (Copy) وتغيير اسم ملف (Rename) . وهكذا .

انظر الشكل ( ٢٧ - ٩ )



شكل ( ٢٧ - ٩ )

والاختيار (Set Drive) يستخدم في تحديد وحدة الأقراص المستخدمة وذلك بإدخال رمز وحدة الأقراص (A, B, C, ...) التي تحتوي على الملفات المطلوب استخدامها .

والاختيار (Copy File) يستخدم في نسخ أي نوع من الملفات . وهو لذلك يختلف عن الاختيار (Copy) في قائمة التنظيم (Organize) . حيث أن الاختيار (Copy) يستخدم في نسخ ملفات قاعدة البيانات فقط ، أي التي تحتوي على الامتداد (.dbf) .

ارجع الى الأمر (COPY FILE) .

والاختيار (Directory) يستخدم في عرض قائمة أسماء الملفات الموجودة في وحدة الأقراص المستخدمة .

ارجع الى الأمر (DIR) .

والاختيار (Rename) يستخدم لتغيير اسم أي ملف بشرط ألا يكون مفتوحاً في هذا الوقت .

ارجع الى الأمر (RENAME) .

والاختيار (Erase) يستخدم في مسح أي ملف من القرص بشرط ألا يكون

مفتوحا في هذا الوقت .

ارجع الى الأمر (ERASE) .

والاختيار (List Structure) يستخدم في عرض تركيب الملف المفتوح . وهذا الاختيار يسمح أيضا بطباعة هذا التركيب (Structure) على الشاشة .

ارجع الى الأمر (LIST STRUCTURE) .

والاختيار (Import) يستخدم في تحويل الملفات التي تمت كتابتها عن طريق برامج أخرى غير برنامج (DBase III+) الى ملفات يمكن استخدامها مع برنامج (DBase III+) .

ارجع الى الأمر (IMPORT) .

والاختيار (Export) يستخدم في تحويل الملفات المكتوبة بواسطة برنامج (DBase III+) الى ملفات يمكن استخدامها بواسطة برامج أخرى .

ارجع الى الأمر (EXPORT) .

## ٨ - الأمر (AVERAGE)

ويستخدم هذا الأمر في حساب القيم المتوسطة للحقول العددية لجموعة من السجلات . والصورة العامة للأمر كالآتي :

```
AVERAGE[<expression list>][<scope>][WHILE <condition>]
[FOR <condition>] [TO <memvar list>]
```

ويلاحظ هنا أن جميع الاختيارات اختيارية أي يمكن كتابتها أو عدم كتابتها . وعند عدم كتابة أي شيء بعد الأمر (AVERAGE) ، فإن ذلك يؤدي إلى حساب القيم المتوسطة لجميع الحقول العددية لجميع سجلات قاعدة البيانات المفتوحة ، وعرض هذه المتوسطات على الشاشة .

أما الاختيارات الموجودة فإنها تحدد الحقول المطلوب حساب متوسطها، كما تحدد المدى (Scope) أي السجلات المطلوب البحث خلالها، كما تحدد الشروط المطلوب

البحث بناء عليها، ثم تحدد أسماء متغيرات الذاكرة التي يتم تخزين القيم المتوسطة فيها .

والاختيار الأول (expression list) يتم من خلاله كتابة أسماء الحقول المطلوب حساب متوسطها .

والاختيار الثاني (scope) يتم من خلاله تحديد مدى السجلات المطلوب البحث خلاله . حيث يمكن البحث خلال كل الملف (ALL) أو خلال السجلات التالية للسجل الذي يقف عنده المؤشر (Rest) و ... وهكذا .

والاختيار الثالث (WHILE <condition>) يؤدي الى البحث عن السجلات التي تحقق شرطاً أو شروطاً معينة .

والاختيار الرابع (FOR <condition>) يؤدي نفس العمل ، أي يبحث عن السجلات التي تحقق شرطاً أو شروطاً معينة . ولكن الاختيار (WHILE) أسرع في الوصول الى السجلات التي تحقق الشروط .

والاختيار الخامس (<memvar list> TO) يؤدي الى انشاء متغيرات الذاكرة (memvar list) التي يتم فيها تخزين المتوسطات المحسوبة ، وذلك بنفس الترتيب الذي يتم به كتابة هذه المتغيرات .

## مثال

للحصول على متوسطات حقل ساعات العمل (w\_hours) وحقل المرتب (salary) لكل ملف قاعدة البيانات يتم كتابة السطر التالي :

```
AVERAGE w_hours, salary TO avg_hr, avg_sal
```

في هذه الحالة يحتوي المتغير (avg\_hr) على متوسط ساعات العمل كما يحتوي المتغير (avg\_sal) على متوسط المرتبات .

## ٩ - الأمر (BROWSE)

ويؤدي هذا الأمر الى عرض شاشة لتعديل بيانات السجلات وإدخال سجلات جديدة . وهو يؤدي نفس العمل الذي يتم عند الدخول في قائمة التحديث (Update) واختيار (Browse) . ولكن كتابته مسن مشسيرة النقطة (Dot Prompt) أو من خلال البرنامج تعطي مرونة أكبر في تحديد الحقول

المطلوب عرضها على الشاشة .

والصورة العامة لهذا الأمر كالآتي :

BROWSE [FIELDS <field list>][LOCK <N>]  
[FREEZE <field>][NOFOLLOW][NOMENU][WIDTH<N>][NOAPPEND]

ويلاحظ هنا أن جميع الاختيارات اختيارية ، أى يمكن كتابة الأمر (BROWSE) دون كتابة أى شيء بعده . وفي هذه الحالة تظهر شاشة تعديل البيانات وتحتوى على بيانات ١٧ سجلا . ويمكن عرض عدد أكبر من السجلات عن طريق الضغط على مفتاح (F1) لإخفاء قائمة المساعدة (Help) الموجودة أعلى الشاشة . وتظهر الحقول على هيئة أعمدة .

وعند الضغط على مفتاح (F10) يظهر عمود القوائم (menu bar) الخاص بالأمر (Browse) . وقد سبق شرح هذه القوائم فى الباب الخاص بتعديل البيانات فى الكتاب الأول . ويمكن ادخال معظم هذه الاختيارات مع الأمر (BROWSE) من خلال الاختيارات الموجودة مع الأمر والتي يمكن تلخيصها كالآتي : انظر الشكل ( ٢٧ - ١٠ )

CURSOR <-- --> Char: + + Field: Home End Pan: ^ ^	UP DOWN Record: ↑ ↓ Page: PgUp PgDn Help: F1	DELETE Char: Del Field: ^Y Record: ^U	Insert Mode: Ins Exit: ^End Abort: Esc Set Options: ^Home
NAME-----	ADDRESS-----	PHONE-----	FATHER-----
MOHAMED AHMED FAHY	12-ain shams	56526756	AHMED
AHMED HASAN	40-ahran-street	6789889	HASAN
WALAA MOSTAFA	CITY	6394588	MOSTAFA
MAYYHAN MOSTAFA	NASR CITY	7428953	MOSTAFA
FATEN KAMAL	NASR CITY	4676787	KAMA HEREK
BROWSE      << >>      Rec: 1/5      C			
View and edit fields.			

شكل ( ٢٧ - ١٠ )

الاختيار (FIELDS <field list>) ويستخدم لتحديد أسماء الحقول المطلوب عرض البيانات الخاصة بها . ويمكن من خلال هذا الاختيار تحديد أى

ترتيب يراد عرض الحقول به . وذلك عن طريق كتابة أسماء الحقول (Field List) بنفس هذا الترتيب .

والاختيار (<N> LOCK) يستخدم لتثبيت عدد من الحقول المتجاورة في يسار الشاشة والتي لا تتحرك عند الضغط على مفتاحي (<---> Ctrl) أو (<---> Ctrl) لعمل زحزحة أفقية (Horizontal Scrolling) ويؤدي هذا إلى عرض الحقول المختلفة من الملف أمام مجموعة من الحقول تكون ظاهرة دائما على الشاشة . ويتم تحديد عدد الحقول المراد تثبيتها عن طريق العدد N .

والاختيار (<field> FREEZE) يستخدم لتجميد حقل معين حتى يصبح هو الحقل الوحيد المسموح بتعديله . وبالرغم من عرض باقي الحقول على الشاشة ، إلا أن المستخدم لا يستطيع تعديل أى حقل آخر غير هذا الحقل .

والاختيار (NOFOLLOW) يستخدم فقط مع الملفات المفهرسة (Indexed) . وهو يؤدي إلى انتقال المؤشر إلى السجل الجديد في حالة تغيير الحقل الفهرسي (Key Field) . وذلك لأن تغيير الحقل الفهرسي يؤدي إلى تغيير ترتيب السجلات في حين يظل المؤشر مكانه .

والاختيار (NOMENU) يستخدم لمنع استخدام عمود الاختيارات (Menu Bar) الخاص بالأمر (BROWSE) .

والاختيار (<N> WIDTH) يستخدم لتحديد عرض الحقل المراد ظهوره بالحروف حيث يمثل العدد N عدد الحروف في الحقل .

والاختيار (NOAPPEND) يستخدم لمنع المستخدم من إضافة أى سجلات جديدة إلى الملف .

## ١٠ - الأمر (CALL)

ويتيح هذا الأمر لخطط البرامج كتابة برامج منفصلة بلغة التجميع (Assembly Language) وتحميلها داخل ذاكرة الحاسب . وهذه البرامج يجب تحويلها أولا إلى الشفرة الثنائية (Binary Code) . ويتم تحميلها في الذاكرة باستخدام الأمر (LOAD) كما يمكن تنفيذها داخل البرنامج المكتوب بواسطة برنامج اسم البرنامج المطلوب تنفيذه . ويتم تحويل البرنامج من برنامج منفذ (Executable) إلى برنامج ثنائي (Binary) عن طريق الأمر (EXE2BIN) وهو أمر من أوامر نظام التشغيل (MS-DOS) .



## ١١ - الأمر (CANCEL)

ويستخدم هذا الأمر لإيقاف تنفيذ البرنامج والعودة إلى مشيرة النقطة . والصورة العامة له كالآتي :

```
CANCEL
```

وهو لا يحتاج إلى معاملات أخرى .

### مثال

لإيقاف تنفيذ البرنامج عند ضغط المستخدم على الحرف X يتم كتابة الأوامر التالية :

```
IF Choice = 'X'
  CANCEL
ENDIF
```

## ١٢ - الأمر (CHANGE)

ويؤدي هذا الأمر إلى عرض شاشة تصحيح مثل الشاشة التي تظهر مع استخدام الاختيار (Edit) من قائمة التحديث (Update) . ولكن كتابة الأمر من مشيرة النقطة (Dot Prompt) أو من خلال البرنامج تتيح التحكم في الحقول المطلوب تعديلها وكذلك تحديد السجل المطلوب تعديله بسهولة . والصورة العامة للأمر كالآتي :

```
CHANGE [<scope>] [FIELDS <field list>]
      [WHILE <condition>] [FOR <condition>]
```

ويلاحظ أن جميع الاختيارات اختيارية ، أي يمكن كتابة الأمر (CHANGE) دون كتابة أي شيء بعده . ويؤدي ذلك إلى ظهور شاشة التصحيح التي يتم من خلالها تعديل السجلات على التوالي سجلاً تلو الآخر بدءاً من السجل الذي يقف عنده مؤشر

السجلات (Record Pointer) . ويتم الانتقال من السجل إلى السجل التالي بالضغط على مفتاح (PgDn) .

أما الاختيارات الموجودة فإنها تحدد الحقول المطلوب تعديلها وكذلك السجل المطلوب تعديله . ويمكن تلخيص هذه الاختيارات كالآتي :

الاختيار (<scope>) يستخدم في تحديد مدى السجلات المطلوب البحث خلالها . وقد يكون البحث خلال كل الملف (ALL) أو بدءا من السجل الذي يقف عند المؤشر وحتى آخر الملف (Rest) . كما يمكن تحديد سجل معين بكتابة رقمه .

والاختيار (FIELDS<field list>) يستخدم لتحديد أسماء الحقول المطلوب تعديلها . وهي الحقول التي تظهر على شاشة الإدخال .

والاختيار (WHILE <condition>) يستخدم للبحث عن السجلات التي تحقق الشرط أو الشروط التي يتم ادخالها كما يستخدم الاختيار (FOR <condition>) لنفس الغرض .

## ملاحظة

الأمر (CHANGE) والأمر (EDIT) متماثلان تماما .

## مثال

لتعديل حقول الاسم والعنوان وتاريخ الميلاد في ملف بيانات الطلبة (Cadets) يمكن كتابة السطور التالية :

```
USE Cadets
CHANGE FIELDS name, address, birth_d
```

## ١٣ - الأمر (CLEAR)

ويستخدم هذا الأمر في مسح الشاشة ووضع مؤشر الشاشة عند أعلى نقطة من اليسار ، وهي النقطة التي أحداثياتها ( صفر ، صفر ) . كما يمكن مسح جزء فقط من الشاشة عن طريق كتابة الأمر (@....CLEAR) مع كتابة الأحداثيات المطلوب

المسح بدءا منها بعد الحرف @ . وذلك كالآتي مثلا :

@ 10,20 CLEAR

#### ١٤ - الأمر (CLEAR ALL)

ويؤدي هذا الأمر الى اغلاق جميع ملفات قواعد البيانات المفتوحة والملفات المرتبطة بها مثل ملفات الفهرس (Index Files) وملفات التشكيل (Format Files) و ..... الخ . كما يؤدي هذا الأمر أيضا الى مسح متغيرات الذاكرة (Memory variables) .

#### ١٥ - الأمر (CLEAR FIELDS)

ويستخدم هذا الأمر في مسح الحقول التي سبق تحصيلها بواسطة الأمر (SET FIELDS TO) . والأمر (SET FIELDS TO) يؤدي الى تحديد الحقول التي يتم استخدامها فقط من الملف ، في حين تبقى باقي الحقول مغلقة وغير مستخدمة . ولذلك يستخدم الأمر (CLEAR FIELDS) في اعادة الملف الى وضعه المبدئي . حيث تصبح جميع الحقول عاملة (Active) .

#### ١٦ - الأمر (CLEAR GETS)

ويستخدم هذا الأمر في مسح كل المتغيرات التي تم انشاؤها بواسطة الأمر (@...GET) . وهذه المتغيرات لا تزيد عن ١٢٨ متغيراً ، وذلك حسب الوضع المبدئي (Default) للبرنامج .

ويفضل دائما عند زيادة هذه المتغيرات أن يتم مسح بعضها بواسطة هذا الأمر ، وذلك بعد تخزينها في متغيرات ذاكرة . خصوصا في الحالات التي يراد فيها انشاء شاشات ادخال من عدة صفحات .

## ١٧ - الأمر (CLEAR MEMORY)

ويستخدم هذا الأمر في مسح متغيرات الذاكرة من الذاكرة المؤقتة للحاسب . وهو يماثل الأمر (RELEASE ALL) ، ولكن الاختلاف بين الأمرين أن الأمر (CLEAR MEMORY) يسمح كل متغيرات الذاكرة سواء كانت عامة (Public) أو خاصة (Private) . أما الأمر (RELEASE ALL) فإنه يسمح للمتغيرات الخاصة فقط .

## ١٨ - الأمر (CLOSE)

ويستخدم هذا الأمر لإغلاق الملفات المفتوحة . وهو يكون على إحدى صورتين :

CLOSE < file type >

أو

CLOSE ALL

والصورة الأولى يتم عن طريقها إغلاق نوع معين من الملفات . أما الصورة الثانية فإنها تستخدم في إغلاق جميع الملفات المفتوحة . ولتوضيح ذلك يمكن كتابة الأوامر التالية :

CLOSE DATABASES  
CLOSE INDEX  
CLOSE FORMAT  
CLOSE PROCEDURE  
CLOSE ALL

## ١٩ - الأمر (CONTINUE)

ويستخدم هذا الأمر مع الأمر (LOCATE) لتوجيه مؤشر السجلات (Record Pointer) إلى السجل المطلوب . حيث يستخدم الأمر (CONTINUE) في تحريك المؤشر إلى السجل الثاني الذي يحقق الشرط ثم السجل الذي يليه ، وهكذا .

ارجع إلى الأمر (LOCATE) .

## ٢٠ - الأمر (COPY)

وهو يشبه الأمر (APPEND FROM) ولكنه لا يضيف سجلات في نهاية الملف المفتوح ، بل ينسخ الملف المفتوح كله أو جزءا منه في ملف آخر جديد . وهناك صورتان لهذا الأمر . الصورة الأولى تستخدم في نسخ ملف قاعدة البيانات كله أو جزء منه في ملف قاعدة بيانات آخر . والصورة الثانية تستخدم في نسخ ملف قاعدة البيانات كله أو جزء منه في ملف آخر لا يشتط أن يكون ملف قاعدة بيانات .

### ١ - الصورة الأولى

```
COPY TO <new filename>[<scope>][FIELDS <fieldlist>]
[ WHILE <condition>] [FOR <condition>]
```

ويلاحظ أن جميع الاختيارات اختيارية ما عدا اسم الملف (<new filename>). وعند كتابة الأمر بدون باقي الاختيارات كالآتي مثلا :

```
COPY TO <new filename>
```

فان هذا يؤدي الى نسخ الملف المفتوح بالكامل في الملف الجديد الذي يتم كتابة اسمه .

أما اذا اريد نسخ حقول معينة أو سجلات معينة فقط فيتم استخدام الاختيارات الأخرى . وهذه الاختيارات تتلخص في الآتي :

الاختيار (<scope>) ويستخدم لتحديد مدى محدد يراد البحث خلاله .

والاختيار (<FIELDS field list>) يستخدم لتحديد حقول معينة فقط يراد نسخها الى الملف الجديد .

والاختيار (<WHILE condition>) يستخدم في البحث عن سجلات تحقق شرطا أو شروطا معينة لنسخها في الملف الجديد . وكذلك يستخدم الاختيار (<FOR condition>) لنفس الغرض .

## ب - الصورة الثانية

`COPY TO <new filename> [TYPE <file type>]`

وتستخدم هذه الصورة في نسخ الملف المفتوح في ملف آخر يمكن استخدامه بواسطة برامج أخرى غير برنامج (Dbase III+) . ويجب في هذه الحالة إضافة الامتداد الخاص بهذا الملف . كما يجب كتابة نوع هذا الملف مكان الاختيار (file type) .

فمثلا لنسخ ملف قاعدة البيانات الى ملف يمكن استخدامه بواسطة برنامج لوتس ٢-٢-١ (Lotus 123) يتم كتابة السطر التالي :

`COPY TO cadets.wks TYPE wks`

كما يمكن نسخ مجموعة من الحقول فقط وكذلك مجموعة من السجلات عن طريق كتابة الاختيارات التي سبق شرحها في الصورة الأولى .

## ٢١ - الأمر (COPY FILE)

يستخدم هذا الأمر لنسخ أى نوع من الملفات . والصورة العامة له كالآتي :

`COPY FILE <file1> TO <file2>`

ويجب ملاحظة أن أسماء الملفات هنا يجب أن تتضمن الامتداد (Extension) ووحدرة الأقراص الموجود عليها كل ملف . كما يجب ملاحظة أن الملفات يجب ألا تكون مفتوحة .

وعند نسخ ملف قاعدة بيانات (Dbase file) يحتوى على حقول ملاحظات (memo) فيجب نسخ ملف الملاحظات المرتبط به . وهو الملف الذى يحتوى على الامتداد (.dbt) .

## ٢٢ - الأمر (COPY STRUCTURE)

يستخدم هذا الأمر في نسخ تركيب ملف قاعدة البيانات فقط دون نسخ السجلات المخزنة به . والصورة العامة له كالآتي :

COPY STRUCTURE TO <filename> [FIELDS <field list>]

واسم الملف يجب أن يشمل وحدة الأقراص الموجود عليها الملف إذا كانت غير وحدة الأقراص البدئية (Default) . ويستخدم الاختيار (FIELDS) في تحديد حقول معينة يراد نسخها في هذا التركيب .

## ٢٢ - الأمر (COPY STRUCTURE EXTENDED)

يستخدم هذا الأمر في نسخ تركيب ملف قاعدة البيانات في ملف آخر يحتوي على أربعة حقول فقط وهي (Field Name) ، (Field Type) ، (Field Length) ، (Decimal numbers) .

ويتم ادخال حقول الملف المنسوخ الى الملف الجديد كسجلات . وتستخدم هذه الطريقة في تصميم البرامج التطبيقية التي يراد السماح للمستخدم بتعديل تركيب ملف قاعدة البيانات من خلالها . دون الحاجة الى استخدام الأمر (MODIFY STRUCTURE) . حيث يتم تعديل تركيب الملف مثل تعديل أى سجل في قاعدة البيانات . والصورة العامة للأمر كالآتي :

COPY TO <new file> STRUCTURE EXTENDED

وبعد السماح للمستخدم بادخال التعديلات المطلوبة على هذا التركيب ، يتم انشاء ملف قاعدة بيانات جديد من هذا التركيب باستخدام الأمر (CREATE FROM) . فمثلا إذا كان هناك ملف للموظفين يحتوي على التركيب التالي :

Field	Field Name	Type	width	Dec
1	name	character	30	
2	address	character	30	
3	phone	character	10	

فعند استخدام الأمر التالي :

COPY TO newname STRUCTURE EXTENDED

يصبح تركيب الملف كالآتي :

FFIELD	Field Name	Type	width	Dec
1	Field_name	character	10	
2	Field_type	character	1	
3	Field_len	Numeric	3	
4	Field_dec	Numeric	3	

وعند عرض سجلات هذا الملف بواسطة الأمر (List) مثلا يظهر الآتي :

Record	Field_Name	Field_Type	Width-Len	Field-dec
1	name	C	30	
2	address	C	30	
3	phone	C	10	

وفي هذه الحالة يمكن تعديل حقول الملف باستخدام أى أمر من أوامر التعديل مثل (Edit) ، (Change) .

## ٢٤ - الأمر (COUNT)

ويستخدم هذا الأمر فى حساب عدد السجلات التى تحقق شرطا أو شروطا معينة . والصورة العامة للأمر كالآتي :

```
COUNT[<Scope>][WHILE<codition>]
[FOR <condition>][TO <memvar>]
```

ويلاحظ أن جميع الاختيارات اختيارية ، حيث يمكن كتابة الأمر (COUNT) دون كتابة أى شيء بعده . وفى هذه الحالة يتم حساب عدد السجلات فى ملف قاعدة البيانات المفتوح .

أما إذا أريد حساب عدد السجلات التى تحقق شروطا معينة ، يتم استخدام الشروط الموجودة مع الأمر . كما يمكن استخدام متغير الذاكرة (memvar) فى تخزين هذا العدد لاستخدامه فى البرنامج حسب الحاجة .

فمثلا يمكن كتابة الأمر التالى :



COUNT FOR name = "Mohamed" TO mname

في هذه الحالة يتم حساب عدد الأشخاص الذين يبدأ اسمهم بالاسم (Mohamed) ثم يتم تخزين هذا العدد في المتغير (mname) .

## ٢٥ - الأمر (CREATE)

يستخدم هذا الأمر في إنشاء ملف قاعدة البيانات . وهو يؤدي الى ظهور الشاشة التي تظهر عند استخدام قائمة الانشاء (Create) في القائمة الرئيسية لبرنامج المساعد (Assistant) واختيار (Database file) .

ارجع الى الجزء الخاص ببرنامج المساعد (Assistant) في الكتاب الاول .

## ٢٦ - الأمر (CREATE FROM)

ويستخدم هذا الأمر في إنشاء ملف قاعدة بيانات من ملف سبق نسخه بواسطة الأمر (COPY STRUCTURE EXTENDED) .

والصورة العامة له كالآتي :

CREATE <new file> FROM <structure extended file >

. ارجع الى الأمر (COPY STRUCTURE EXTENDED) .

## ٢٧ - الأمر (CREATE/MODIFY LABEL)

يستخدم هذا الأمر في إنشاء العناوين المختصرة (Labels) ، وهي صورة مصفوفة من التقارير (Reports) تعطي معلومات سريعة عن سجلات قاعدة البيانات . والصورة العامة للأمر كالآتي :

CREATE/MODIFY LABEL <filename>/?

وهذا الأمر يؤدي الى ظهور نفس الشاشة التي تظهر عند الدخول في قائمة الانشاء (Create) أو قائمة التعديل (Modify) واختيار (Label) ، مع ملاحظة

أن الأمر (CREATE) يستخدم في إنشاء التقرير والأمر (MODIFY) يستخدم في إنشائه أو تعديله بعد ذلك . ويتم كتابة اسم الملف بدون الامتداد ، حيث أن البرنامج يضيف الامتداد (.lbl) اليه ألياً . وإذا لم يتذكر المستخدم اسم الملف المطلوب فإنه يكتب (?) بدلا من اسم الملف . وفي هذه الحالة تظهر قائمة بأسماء ملفات التقارير المختصرة (Labels) الموجودة في القرص أو في الكتالوج المفتوح .

ويحتوي عمود الاختيارات (Menu bar) الذي يظهر عند كتابة هذا الأمر على ثلاثة قوائم يمكن تلخيصها كالآتي :

### ١- قائمة الاختيارات (Options)

وهي تحدد حجم التقرير (Size) كما هو موضح في شكل ( ٢٧ - ١١ ) ، وهناك خمسة أحجام قياسية يظهر أحدها على أول سطر في القائمة وتظهر باقي الأحجام عند الضغط على مفتاح الإدخال . كما تستخدم باقي اختيارات القائمة في تحديد عرض التقرير وارتفاعه والهوامش الأيسر والمسافة بين التقارير ... وهكذا .

Options		Contents		Exit	
Predefined size: 3 1/2 x 15/16 by 1					
Label width:		35			
Label height:		5			
Left margin:		0			
Lines between labels:		1			
Spaces between labels:		0			
Labels across page:		1			
CURSOR ← →		UP	DOWN	DELETE	Insert Mode: ins
Char: ← →		Record: ↑ ↓		Char: Del	Exit: ^End
Field: Home End		Page: PgUp PgDn		Field: ^Y	Abort: Esc
Pan: ^* ^)		Help: F1		Record: ^H	Set Options: ^None
CREATE LABEL    <<C>> C:M.LBL    Out: 1/2					

الشكل ( ٢٧ - ١١ )

## ٢ - قائمة المحتويات (Contents)

ويتم عن طريقها تحديد محتويات التقرير ، أى أسماء الحقول أو متغيرات الذاكرة المثلة لها . وعند كتابة أسماء الحقول يمكن الضغط على مفتاح (F10) لعرض قائمة بأسماء الحقول واختيار الحقول المراد عرضها في التقرير . ويمكن عرض أى عدد من السطور في التقرير يحتوى كل منها على حقل أو أكثر . وعندما يراد عرض أكثر من حقل فى نفس السطر يتم فصلها بواسطة الفاصلة ( , ) .

## ٣ - قائمة الخروج (Exit)

ويستخدم هذه القائمة فى الخروج من قوائم التقارير المختصرة (Label) وهى تحتوى على اختيارين :

١ - الاختيار (Save) ويستخدم فى تخزين التقرير الذى تم انشاؤه أو التعديلات التى تم ادخالها على تقرير سابق . ويمكن تنفيذ هذه العملية أيضا بالضغط على مفتاحي (Ctrl-End) .

ب- الاختيار (Abandon) ويستخدم فى الخروج من قوائم التقارير المختصرة دون تخزين التقرير أو التعديلات التى تم ادخالها على تقرير سابق .

## ٢٨ - الأمر (CREATE/MODIFY QUERY)

يستخدم هذا الأمر فى انشاء أو تعديل مرشح (Filter) لاستخدامه مع قاعدة البيانات . وهذا المرشح يؤدي نفس عمل المرشح بمفهومة الميكانيكي . حيث أنه لايسمح بالمرور الا للسجلات التى تحقق الشرط أو الشروط المطلوبة . والصورة العامة للأمر كالآتى :

CREATE/MODIFY QUERY <filename>/?

ويستخدم الأمر (CREATE) لانشاء ملف مرشح جديد . كما يستخدم الأمر (MODIFY) لتعديل ملف سبق انشاؤه . ويتم كتابة اسم الملف بدون الامتداد حيث أن البرنامج يضيفه آليا . وإذا لم يتذكر المستخدم اسم الملف المطلوب يمكنه كتابة الحرف (?) لعرض أسماء ملفات البحث المخزنة على القرص أو فى الكتالوج لاختيار الملف المطلوب تعديله .

وهذا الأمر يؤدي إلى ظهور نفس الشاشة التي تظهر عند الدخول في قائمة الانشاء (Create) أو قائمة التعديل (Modify) واختيار (Query) . حيث يظهر عمود الاختيارات (Menu bar) الذي يحتوي على القوائم التالية :

انظر شكل ( ٢٧ - ١٢ )

Set Filter	Nest	Display	Exit
<div> Field Name  Operator  Constant/Expression  Connect  Line Number      1 </div>			
Line	Field	Operator	Constant/Expression
1			
2			
3			
4			
5			
6			
7			
CREATE QUERY       <C> C:M.QRY       Opt: 1/2			

شكل ( ٢٧ - ١٢ )

#### ١ - القائمة (Set Filter)

وهذه القائمة تستخدم في تحديد الشروط المطلوب ادخالها في ملف البحث حيث تظهر عدة اختيارات تساعد المستخدم على ادخال هذه الشروط في الجدول الذي يظهر على الشاشة ، وهذه الاختيارات تكون كالآتي :

- الاختيار الأول هو اسم الحقل (Field Name) الذي يؤدي الى عرض قائمة بأسماء الحقول لاختيار الحقل المطلوب ادخاله في الشرط .
- الاختيار الثاني هو معاميل المقارنة (Operator) ويؤدي الى ظهور قائمة بالمعاملات التي يتم اختيار معاميل المقارنة المطلوب منها .
- الاختيار الثالث هو القيمة المطلوب مقارنتها (Constant/Expression) . وهو يتيح للمستخدم كتابة القيمة المطلوب مقارنة محتويات الحقل الذي تم اختياره بها .

- الاختيار الرابع هو الربط (Connect) وهو يساعد المستخدم على الربط بين عدة شروط باستخدام للعوامل المنطقية مثل (AND) ، (OR) .
  - الاختيار الخامس هو رقم السطر (Line Number) وهو يساعد المستخدم على اختيار رقم سطر معين في الجدول لتعديله . ويتغير هذا الرقم ألياً عند الانتقال من سطر إلى آخر .
- ويمكن للمستخدم إضافة أى سطر بين السطور التي تمت كتابتها في الجدول بالضغط على مفتاحي (Ctrl-N) . كما يمكنه أيضاً مسح أى سطر من السطور بالضغط على مفتاحي (Ctrl-U) .

## ٢ - القائمة (Nest)

هذه القائمة تستخدم في وضع الأقواس حول الشروط التي يراد وضعها داخل الأقواس . وذلك للتحكم في ترتيب تنفيذ الشروط المختلفة . وهي تتيح للمستخدم إضافة أقواس أو حذف أقواس سبق إضافتها في أى سطر من سطور الجدول . ويفيد ذلك في تكوين العلاقات المركبة التي تتكون من عدة شروط .

## ٣ - القائمة (Display)

وتستخدم هذه القائمة في عرض السجلات التي تحقق الشروط التي سبق ادخالها . وذلك للتأكد أن الشروط قد تمت كتابتها بدقة . حيث يتم عرض بيانات أول سجل يحقق الشرط . ثم يتم الانتقال إلى السجلات التالية عن طريق الضغط على مفتاح (PgDn) .

## ٤ - قائمة الخروج (EXIT)

وهي تستخدم لتخزين الملف الذي تم تكوينه أو الخروج دون تخزين الملف . أرجع إلى الجزء الخاص بإنشاء ملفات البحث عن طريق برنامج المساعد (Assistant)

## ٢٩ - الأمر (CREATE /MODIFY REPORT)

يستخدم هذا الأمر في إنشاء التقارير (Reports) الجديدة أو تعديل تقارير سبق إنشاؤها . وهذه التقارير يتم تصميمها على هيئة أعمدة تمثل الحقول المختلفة . والصورة العامة للأمر كالآتي :

CREATE/MODIFY REPORT <filename>/?

وهذا الأمر يؤدي إلى ظهور نفس الشاشة التي تظهر عند الدخول في قائمة الانشاء (CREATE) واختيار (REPORT) . ويستخدم الأمر (CREATE) لانشاء ملف تقارير جديد في حين يستخدم الأمر (MODIFY) في انشاء أو تعديل ملف تقرير سبق انشاؤه . ويتم ادخال اسم الملف بدون الامتداد حيث أن البرنامج يضيف الامتداد (.FRM) ألياً . وإذا لم يتذكر المستخدم اسم الملف المطلوب تعديله فإنه يكتب الحرف (?) لعرض جميع الملفات المخزنة على القرص أو في الكتالوج المفتوح ، ثم اختيار الملف المطلوب .

وعند كتابة هذا الأمر يظهر عمود الاختيارات (Menu Bar) الذي يحتوي على مجموعة من الاختيارات يمكن تلخيصها كالآتي :

### ١- الاختيارات (Options)

وهي تحتوي على عدة اختيارات يتم عن طريقها تحديد عنوان للصفحة (Page Title) بالإضافة إلى الاختيارات الخاصة بأبعاد الصفحة مثل العرض والهوامش وعدد السطور في الصفحة و ... الخ .

وعند ادخال عنوان الصفحة (Page Title) ، فإن البرنامج يتيح للمستخدم ادخال حتي (٤) سطور بحد أقصى (٦٠) حرفاً . أما باقي الاختيارات في القائمة فيمكن تعديلها حسب الحاجة ولكن في الغالب تكون القيم المبدئية (Default) المكتوبة أمامها مناسبة .

انظر الشكل ( ٢٧ - ١٢ )

Options	Groups	Columns	Locate	Exit
Page title				
Page width (positions)	80			
Left margin	8			
Right margin	0			
Lines per page	58			
Double space report	No			
Page eject before printing	Yes			
Page eject after printing	No			
Plain page	No			

CURSOR (— —)	UP DOWN	DELETE	Insert Mode: Ins
Char: < >	Record: ↑ ↓	Char: Del	Exit: ^End
Field: Home End	Page: PgUp PgDn	Field: ^Y	Abort: Esc
Par: ^← ^→	Help: F1	Record: ^U	Set Options: ^Home

CREATE REPORT	<G:>  G:R.FRM	Opt: 1/9
---------------	---------------	----------

شكل ( ٢٧ - ١٢ )

## ب - المجموعات (Groups)

استخدام المجموعات اختياري ، وبعض التقارير لا تحتاج الى استخدام هذا الاختيار . ولكنها في احيان أخرى تكون مطلوبة لتحسين شكل التقرير والحصول على معلومات أكثر تفصيلا . حيث أن هذه المجموعات تتيح للمستخدم تقسيم بيانات التقرير بناء على البيانات الموجودة في أحد الحقول . فمثلا إذا كانت هناك قاعدة بيانات للموظفين الذين يختلفون في الوظائف (Jobs) . فيمكن تصميم التقرير مع استخدام حقل الوظيفة (Job) كأساس للتقسيم ، وذلك عن طريق كتابة اسم هذا الحقل مكان الاختيار (Group on Expression) وعند طباعة التقرير تظهر بيانات الموظفين الذين يشتركون في الوظيفة في مجموعة واحدة .

وإذا كانت هناك حقول عديدة في ملف قاعدة البيانات ، فيتم تجميع هذه البيانات لكل مجموعة على حدة .

وهناك عدة اختيارات تحت قائمة المجموعات وهي كالآتي :

- الاختيار (Group on expression) وهو يستخدم لإدخال اسم الحقل المراد التقسيم بناءً عليه .
- الاختيار (Group Heading) ويستخدم لإدخال عنوان لكل مجموعة يوضح الحقل الذي تم التقسيم بناءً عليه . فمثلاً يمكن كتابة (Client's Job) كعنوان لكل مجموعة .
- الاختيار (Summary Report Only) وهو يستخدم لتحديد ما إذا كان المطلوب الحصول على تقرير مختصر أو تقرير تفصيلي . وفي حالة التقرير المختصر يتم عرض البيانات العددية لكل مجموعة بعد تجميعها دون عرض باقي البيانات . أما التقرير التفصيلي فيتضمن جميع بيانات السجلات الموجودة في كل مجموعة .
- الاختيار (Page Eject After Group) ويستخدم لكتابة بيانات كل مجموعة في صفحة منفصلة .
- الاختيار (Sub-group on Expression) ويستخدم لتجميع البيانات في مجموعات فرعية داخل المجموعات . ويتم ذلك بناءً على حقل آخر يتم اختياره .
- الاختيار (Sub-group Heading) ويستخدم لتحديد عنوان لكل مجموعة فرعية . انظر شكل ( ٢٧ - ١٤ )

Options	<b>Groups</b>	Columns	Locate	Exit	07:32																
<div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 80%;"> <p>Group on expression</p> <p>Group heading</p> <p>Summary report only      No</p> <p>Page eject after group</p> <p>Sub-group on expression</p> <p>Sub-group heading</p> </div>																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">CURSOR    (← →)</td> <td style="width: 25%;">UP      DOWN</td> <td style="width: 25%;">DELETE</td> <td style="width: 25%;">Insert Mode: Ins</td> </tr> <tr> <td>Char:    ← →</td> <td>Record: ↑ ↓</td> <td>Char:    Del</td> <td>Exit:    ^End</td> </tr> <tr> <td>Field: Home End</td> <td>Page: PgUp PgDn</td> <td>Field: ^V</td> <td>Abort:    Esc</td> </tr> <tr> <td>Pan:    ^← ^→</td> <td>Help:    F1</td> <td>Record: ^U</td> <td>Set Options: ^Home</td> </tr> </table>						CURSOR    (← →)	UP      DOWN	DELETE	Insert Mode: Ins	Char:    ← →	Record: ↑ ↓	Char:    Del	Exit:    ^End	Field: Home End	Page: PgUp PgDn	Field: ^V	Abort:    Esc	Pan:    ^← ^→	Help:    F1	Record: ^U	Set Options: ^Home
CURSOR    (← →)	UP      DOWN	DELETE	Insert Mode: Ins																		
Char:    ← →	Record: ↑ ↓	Char:    Del	Exit:    ^End																		
Field: Home End	Page: PgUp PgDn	Field: ^V	Abort:    Esc																		
Pan:    ^← ^→	Help:    F1	Record: ^U	Set Options: ^Home																		
<p><b>CREATE REPORT</b>    [KC:]    C:R.FRN    Out: 1/3</p> <p>Position selection bar - ↑↓. Select - ←→. Leave menu - ^←.</p> <p>Enter a field or expression on which to break for the first level of subgr</p>																					

شكل ( ٢٧ - ١٤ )



ح - الأعمدة (Columns)

وهذه القائمة تستخدم في تحديد البيانات المطلوب كتابتها في الأعمدة الخاصة بالتقرير ، مع تحديد مكان هذه البيانات وشكلها في التقرير . وتحتوي هذه القائمة كما هو موضح في شكل ( ٢٧ - ١٥ ) على الاختبارات التالية :

```
Options      Groups      Column#      Locate      Exit      MZ:48:
-----
Contents      name
Heading      name
Width      30
Decimal places
Total this column

Report Format
>>>>>>name
```

=====

REPORT REPORT [K<:] C.R.FRM Column: 1  
Position selection has - tl Select - d Prev/Next column = Page/End

شكل ( ٢٧ - ١٥ )

- الاختيار (Contents) ويستخدم في تحديد محتويات كل عمود من أعمدة التقرير . ويمكن أن يحتوى العمود على حقل بيانات (Field) أو متغير ذاكرة (Memory Variable) أو علاقة رياضية بين أكثر من حقل .
- الاختيار (Heading) ويستخدم هذا الاختيار لكتابة عنوان للعمود .
- الاختيار (Width) ويستخدم هذا الاختيار لتحديد عرض العمود . وهو يكون نفس عرض الحقل الذي يتم ادخاله في هذا العمود . وإذا كان العنوان (Heading) أطول من عرض الحقل ، فإن عرض العمود يمتد ليعطي عرض العنوان . كما يمكن زيادة أو انقاص عرض العمود بكتابة العرض المطلوب . وعند كتابة عرض أقل من عرض الحقل ، فإن البيانات التي تظهر في هذا العمود ينتقل جزء منها الى السطر التالي .

- الاختيار (Decimal Places) يستخدم مع الحقول العددية فقط . وهو يحدد عدد الكسور العشرية المطلوب ظهورها في التقرير لهذا العدد ، وهو يأخذ نفس القيمة التي سبق ادخالها عند انشاء ملف قاعدة البيانات . كما يمكن ادخال أى قيمة أخرى .
- الاختيار (Total This Column) يستخدم مع الحقول العددية عندما يراد تجميع البيانات العددية الخاصة بهذا الحقل لمجموعة السجلات التي تظهر في التقرير . والقيمة المبدئية لهذا الاختيار هي (Yes) ، وإذا أريد تغييرها يتم كتابة (N) أمام هذا الاختيار . وإذا أريد مسح العمود بعد ادخاله يتم الضغط على مفتاحي (Ctrl-U) . وإذا أريد اضافة عمود قبل العمود الجارى ادخاله يتم الضغط على مفتاحي (Ctrl-N) .

#### د- الاختيار : (Locate)

يستخدم للوصول الى أى عمود سبق ادخال بياناته . حيث تظهر قائمة بأسماء الأعمدة التي سبق ادخالها ، فيتم اختيار العمود المطلوب تعديله .

انظر الشكل ( ٢٧ - ١٦ )

Options	Groups	Columns	Locate	Exit	28:13:58 pm
			<div style="border: 1px solid black; padding: 2px;">           Name:            NAME         </div>		
Report Format >>>>>>>name					
-----					
-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX##					
CREATE REPORT    <<C:>C:R.FRM    Opt: 1/2					

شكل ( ٢٧ - ١٦ )

## هـ - الاختيار : (Exit)

- يستخدم لتخزين التقرير الذي تم انشاؤه أو التعديلات التي تم ادخالها .  
أو الخروج دون تخزين التقرير .
- ارجع الى الجزء الخاص بانشاء التقرير في برنامج المساعد (Assistant)  
في الكتاب الاول .

## ٢٠ - الأمر (CREATE/MODIFY SCREEN)

يستخدم هذا الأمر في انشاء شاشات الادخال الجديدة أو تعديل شاشات سبق انشاؤها . وهو يؤدي الى انشاء ملفين أحدهما بالامتداد (.scr) والآخر بالامتداد (.fmt) . حيث يساعد الملف ذو الامتداد (.scr) على تعديل شاشة الادخال ثم تخزينها في الملف ذي الامتداد (.fmt) .

والصورة العامة للأمر كالآتي :

CREATE/MODIFY SCREEN <filename>/?

ويستخدم الأمر (CREATE) في انشاء ملف تشكيل (Format File) جديد . كما يستخدم الأمر (MODIFY) في انشاء أو تعديل ملف تشكيل سبق ادخاله . ويكتب اسم الملف بدون الامتداد ، حيث أن البرنامج يضيف اليه الامتداد (.scr) . وإذا لم يتذكر المستخدم اسم ملف التشكيل المطلوب تعديله يمكنه كتابة (?) بدلا من اسم الملف لعرض جميع ملفات التشكيل المخزنة على القرص أو في الكتالوج المفتوح ، حيث يتم اختيار الملف المطلوب تعديله . وعند استخدام هذا الأمر تظهر الشاشة التي تظهر عند الدخول في قائمة الانشاء واختيار (Format For Screen) .

### تحذير

في حالة مسح الملف ذي الامتداد (.scr) لايمكن تعديل ملف التشكيل ذي الامتداد (.fmt) المقابل له . حيث أن ملف التشكيل ذي الامتداد (.fmt) لايمكن ترجمته الى ملف الشاشة ذي الامتداد (.scr) . ولكن في هذه الحالة يمكن تعديله كملف برنامج باستخدام الأمر (MODIFY COMMAND) . وعند تعديل ملف التشكيل ذي الامتداد (.fmt) بواسطة الأمر (MODIFY COMMAND) ، فلا يمكن تعديله بعد ذلك باستخدام الأمر (MODIFY SCREEN) .

ولمزيد من المعلومات عن هذا الأمر أرجع إلى الجزء الخاص بإنشاء شاشة ادخال البيانات من خلال برنامج المساعد (Assistant) في الكتاب الأول .

### ٣١ - الأمر (CREATE/MODIFY VIEW)

يستخدم هذا الأمر في ربط عدة ملفات قواعد بيانات عن طريق ملف واحد يسمى ملف المنظر (View File) . وهذا الملف يحتوي على حقول يتم اختيارها من عدة ملفات . ويستخدم هذا الملف الجديد في عرض أى بيانات من هذه الملفات أو تعديلها .

والصورة العامة للأمر كالآتي :

CREATE/MODIFY VIEW <filename>/?

ويستخدم الأمر (CREATE) في إنشاء ملف المنظر (View File) كما يستخدم الأمر (MODIFY) في إنشاء أو تعديل أى ملف منظر سبق إنشاؤه . ويتم كتابة اسم الملف بدون الامتداد حيث أن البرنامج يضيف الامتداد (.VUE) أليها . وإذا لم يتذكر المستخدم اسم الملف المطلوب تعديله يمكنه كتابة الحرف (?) مكان اسم الملف فتظهر قائمة بأسماء ملفات المنظر الموجودة على القرص أو في الكتالوج المفتوح .

وعند تخزين ملف المنظر الذي تم إنشاؤه يصبح الملف مفتوحا . كما يمكن فتح أى ملف منظر بكتابة الأمر (SET VIEW TO <filename>) . كما يمكن إغلاق أى ملف منظر مفتوح بكتابة الأمر (CLOSE DATABASES) وهذا يؤدي إلى إغلاق جميع ملفات قواعد البيانات والملفات المرتبطة بها مثل ملفات الفهرس والتشكيل . الخ . كما يؤدي إلى إغلاق ملف المنظر الذي يربط هذه الملفات .

وعند كتابة هذا الأمر يظهر عمود الاختيارات (Menu Bar) الذي يحتوي على الاختيارات التالية :

### ١ - الاختيار (Set Up)

وهذا الاختيار يعرض على الشاشة قائمة بأسماء ملفات قواعد البيانات الموجودة على القرص أو في الكتالوج المفتوح حتى يستطيع المستخدم اختيار الملفات المطلوب ادخالها في ملف المنظر . واختيار أى ملف من الملفات يتم تحريك العمود الضوئي (Highlight) إلى اسم هذا الملف ثم الضغط على

مفتاح الإدخال . وهذا يؤدي إلى ظهور مثلث يسار اسم الملف . كما يمكن إزالة هذا المثلث بالضغط على مفتاح الإدخال مرة ثانية . وهذا يعني عدم اختيار هذا الملف . وكل ملف يتم اختياره يتم اختيار ملف الفهرس الخاص به . ويمكن إدخال حتى تسعة ملفات قواعد بيانات والملفات الملحق بها في ملف منظر واحد .

## ٢ - الاختيار (RELATE)

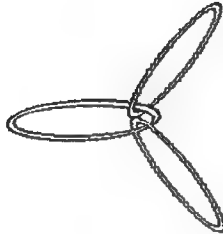
ويستخدم في تحديد الحقول التي يتم عن طريقها ربط الملفات ببعضها . ويتم عن طريق هذه القائمة عرض الملفات التي سبق اختيارها حتى يتم فتح الملفات التي سوف تستخدم في تكوين سلسلة العلاقات (Relation Chain) . وترتيب فتح هذه الملفات مهم جدا . حيث أن أول ملف يتم فتحه يعتبر بداية السلسلة (Chain) .

وبعد فتح هذا الملف تظهر قائمة بباقي الملفات التي سبق اختيارها من خلال قائمة التجهيز (Set Up) فيتم تحديد الملف المطلوب ربطه بالملف الأول . ثم يسأل البرنامج عن اسم الحقل الذي يتم الربط بناء عليه فيتم اختيار اسم هذا الحقل . وبنفس الطريقة يمكن ربط باقي الملفات عندما يراود ربط أكثر من ملفين في نفس السلسلة (Chain) . ويجب ملاحظة أن الربط يتم على التوالي . فلا يجوز مثلا ربط ملف جديد بملف سبق ربطه بملفات تالية . أي أن السلسلة يجب أن تكون على الشكل التالي:



شكل ( ٢٧ - ١٧ )

وليست كالشكل التالي :



شكل ( ٢٧ - ١٨ )

ويمكن ربط الملفين من خلال مشيرة النقطة (Dot Prompt) أو من خلال البرنامج عن طريق كتابة السطور التالية :

```
SELECT 1
USE file1
SELECT 2
USE file2
SELECT 1
SET RELATION TO field1 INTO file2
```

حيث (field1) هو الحقل المشترك بين الملفين (file1) ،  
(file2).

### ٣ - الاختيار (Set Fields)

وهذا الاختيار يسمح للمستخدم باختيار أسماء الحقول المطلوب ادخالها في ملف المنتظر . وفي هذه الحالة يتم اختيار كل ملف من الملفات المرتبطة في العلاقة فتظهر أسماء الحقول الخاصة بهذا الملف . ويتم اختيار جميع الحقول المطلوبة . ويلاحظ أن الحقول تظهر في البداية وأمامها علامة ( ▶ ) دليل على اختيارها . ولإلغاء اختيار أي حقل يتم الضغط على مفتاح الإدخال عند وقوف

المؤشر على اسم هذا الحقل .

#### ٤ - الاختيار (Options)

وهي قائمة تحتوي على الاختيار (Format) الذى يستخدم فى تعديل ملف التشكيل (Format File) الذى سبق انشاؤه لملف المنظر . وإذا أريد انشاء ملف تشكيل جديد يتم انشاؤه من خلال قائمة الانشاء (Create) كما سبق الايضاح .

وتحتوى هذه القائمة أيضا على الاختيار (Filter) . وهذا الاختيار يتيح للمستخدم ادخال الشروط المطلوب ادخالها للبحث عن السجلات .

#### ٥ - الاختيار (Exit)

وهو يتيح للمستخدم تخزين ملف المنظر الذى تم انشاؤه من خلال الاختيار (Save) أو الخروج دون تخزينه (Abandon) .

### ٢٢ - الأمر (CREATE VIEW FROM ENVIRONMENT)

يستخدم هذا الأمر فى انشاء ملف منظر (View File) من العلاقات (Relations) التى سبق انشاؤها من خلال أوامر البرنامج . ويفيد هذا الأمر فى ربط الملفات وإنشاء ملف منظر لها من خلال البرنامج دون الرجوع الى برنامج المساعد (Assistant) والصورة العامة لهذا الأمر كالآتى :

CREATE VIEW <filename> FROM ENVIRONMENT

وعند كتابة هذا الأمر يتم تخزين جميع ملفات قواعد البيانات وملفات الفهرس والعلاقات التى تم انشاؤها بينها عن طريق الأمر (SET RELATION TO) ، كما يتم ادخال جميع الحقول الخاصة بهذه الملفات فى ملف المنظر ، وذلك فى حالة عدم كتابة الأمر (SET FIELDS TO) الذى يمكن من خلاله تحديد حقول معينة .

## ٣٣ - الأمر (DELETE)

يستخدم هذا الأمر في تحديد السجلات المطلوب مسحها . وهو يؤدي نفس العمل الذي يتم تنفيذه في حالة استخدام برنامج المساعد (Assistant) والدخول في قائمة التحديث (Update) واختيار (Delete) . والصورة العامة للأمر كالآتي :

DELETE [<scope>] [WHILE <condition>] [FOR <condition>]

ويلاحظ هنا أن جميع الاختيارات اختيارية . أي يمكن كتابة الأمر دون كتابة أي شيء بعده . وفي هذه الحالة يتم مسح السجل الذي يقف عنده مؤشر السجلات (Record Pointer) في هذا الوقت . ويجب ملاحظة أن هذا الأمر لا يمسح السجلات مباشرة ولكنه يضع علامة عندها حتى يتم مسحها نهائياً باستخدام الأمر (PACK) . ويلاحظ عند عرض السجلات باستخدام الأمر (DISPLAY) والأمر (LIST) ، ظهور علامة (\*) أمام كل سجل سبق مسحه باستخدام الأمر (DELETE) وهذا يعني أن السجل موجود ولكنه جاهز للمسح .

والاختيار (scope) يستخدم لتحديد المدى الذي يتم البحث خلاله عن السجلات المطلوب مسحها . كما يستخدم الاختيار (WHILE <condition>) لتحديد الشرط أو الشروط التي يتم البحث بناء عليها . وكذلك يستخدم الاختيار (FOR <condition>) لنفس الغرض .

مثال :

لتجهيز أول سجل في قاعدة البيانات (Cadets) للمسح يتم كتابة الأوامر التالية :

```
. USE Cadets
. DELETE
1 record deleted
```

يلاحظ ظهور الرسالة الدالة على تجهيز سجل واحد للمسح . كما يمكن كتابة السطر التالي لتجهيز السجل رقم ٨ للمسح .

```
. DELETE RECORD 8
1 record deleted
```

ويجب ملاحظة أن (8 RECORD) هنا تمثل المدى (Scope) .



كما يمكن كتابة السطر التالي لمسح جميع السجلات التي يبدأ حقل الاسم فيها بالاسم (Mohamed) مثلا :

```
DELETE ALL FOR name = 'Mohamed'
```

ويجب ملاحظة أن (ALL) هنا تمثل المدى (scope) ، حيث يتم البحث خلال ملف قاعدة البيانات كله .

## ٢٤ - الأمر (DIR)

يستخدم هذا الأمر في عرض دليل ملفات قواعد البيانات الموجودة على وحدة الأقراص المستخدمة أو الفهرس الفرعي المستخدم . والصورة العامة له كالآتي :

```
DIR [<drive:>] [<path>]
```

وكتابة الأمر دون كتابة أي شيء بعده تؤدي الى ظهور أسماء ملفات قواعد البيانات الموجودة على وحدة الأقراص الحالية أو الفهرس الفرعي الحالي . كما يمكن استخدام هذا الأمر في عرض جميع الملفات عن طريق اضافة الحروف الشاملة (Global Characters) وذلك كالآتي مثلا :

```
DIR *.*
```

وهذا يؤدي الى عرض جميع الملفات الموجودة في وحدة الأقراص أو الفهرس الحالي سواء كانت ملفات قواعد بيانات أو أي ملفات أخرى .

كما يمكن عرض ملفات الفهرس فقط عن طريق كتابة السطر التالي :

```
DIR *.NDX
```

كما يمكن استخدام المسار (Path) كالآتي مثلا :

```
DIR SALES \*.*
```

ويؤدي هذا الى عرض جميع الملفات الموجودة في الفهرس الفرعي (SALES) .

كما يمكن عرض ملفات قواعد البيانات الموجودة في هذا المسار كالآتي :

DIR SALES\

### ٣٥ - الأمر (DISPLAY)

ويستخدم هذا الأمر في عرض بيانات السجلات المطلوبة . والصورة العامة له كالآتي :

```
DISPLAY [<scope>][<expression list>]
[WHILE <condition>][FOR <condition>][ OFF][ TO PRINT]
```

ويلاحظ هنا أن جميع الاختيارات اختيارية حيث يمكن كتابة الأمر دون كتابة أي شيء بعده . وفي هذه الحالة تظهر بيانات السجل الذي يقف عنده مؤشر السجلات (Record Pointer) . كما يتم عرض جميع الحقول الخاصة بهذا السجل .

وعند زيادة بيانات الحقول عن عرض الشاشة تنتقل البيانات إلى السطر التالي . وعند عرض عدد من السجلات يزيد من طول الشاشة ، فإن عرض السجلات يتوقف مؤقتا (Pause) حتى يضغط المستخدم على أي مفتاح لاستكمال العرض .

والاختيار <scope> يستخدم في تحديد المدى المطلوب البحث خلاله .

والاختيار <expression list> يستخدم في تحديد الحقول المطلوب عرض بياناتها .

والاختيار (WHILE <condition>) يستخدم في تحديد السجلات المطلوب عرض بياناتها والتي تحقق الشرط أو الشروط المكتوبة . وكذلك الاختيار (FOR <condition>) يؤدي نفس الغرض . ولكن الاختيار (WHILE) أقوى وأسرع في الوصول إلى السجلات المطلوبة .

والاختيار (OFF) يستخدم عندما يراد عدم ظهور أرقام السجلات على الشاشة .

والاختيار (TO PRINT) يستخدم عندما يراد طباعة البيانات المعروضة ويجب ملاحظة أن أسماء الحقول تظهر فوق البيانات للمعرضة . وعندما يراد عدم عرض أسماء الحقول يستخدم الأمر (SET HEADING OFF) . كما أن أسماء الحقول تظهر بالحالة التي يتم كتابتها بها في الأمر ، أي إذا تمت كتابتها بحروف كبيرة (Capital) تظهر على الشاشة بحروف كبيرة أيضا .

فمثلا لعرض حقول الاسم والعنوان في ملف الطلبة (Cadets) . يتم كتابة السطور التالية :

USE CADETS  
DISPLAY NEXT 3 NAME, ADDRESS :

وفي هذه الحالة تظهر الشاشة التالية :

Record #	NAME	ADDRESS
1	MOHAMED	12 - Ahram Street
2	TAREK	8 - Tayaran Street
3	GALAL	20 - Gomhorya Square

ويلاحظ هنا أن أسماء الحقول في رأس القائمة (Heading) مكتوبة بحروف كبيرة لأنها تم كتابتها في الأمر بحروف كبيرة (Capital) .

### ملاحظة

حقول الملاحظات (memo fields) لا تظهر محتوياتها على الشاشة الا عند كتابتها منفصلة في الأمر . وفي هذه الحالة تظهر محتويات هذا الحقل فقط على الشاشة . فمثلا اذا كان هناك حقل ملاحظات يسمى (Notes) ، فلكي يتم عرض البيانات المخزنة فيه بالنسبة للسجل الذي يقف عنده مؤشر السجلات (Record Pointer) يتم كتابة الأمر التالي :

DISPLAY Notes

وفي هذه الحالة تظهر الملاحظات الخاصة بهذا السجل بعرض (٥٠) حرفا في السطر . واذا اريد تغيير هذا العرض يستخدم الأمر (SET MEMOWIDTH) في تحديد طول السطر في حقل الملاحظات .

### ٣٦ - الأمر (DISPLAY HISTORY)

يستخدم هذا الأمر في عرض آخر عشرين أمرا تم تنفيذها . والصورة العامة له

كالاتي :

## DISPLAY HISTORY [LAST <N>] [TO PRINT]

والاختيار (LAST<N>) يستخدم لتحديد عدد الأوامر المطلوب عرضها على الشاشة . وهذا العدد لا يزيد عن عشرين أمرا حيث أنه هو العدد المبدئي (Default). كما يمكن زيادة هذا العدد عن طريق الأمر (SET HISTORY TO) .

والاختيار (TO PRINT) يستخدم عندما يراد طباعة هذه الأوامر على الطابعة .

## ٣٧ - الأمر (DISPLAY MEMORY)

يستخدم هذا الأمر في عرض حالة متغيرات الذاكرة (Memory Variables) الموجودة في الذاكرة المؤقتة (RAM) في أي وقت . حيث يعرض اسم كل متغير ونوعه وحجمه . كما يعرض أيضا بيانات من حجم الذاكرة المستخدمة وحجم الذاكرة المتاحة . والصورة العامة لهذا الأمر كالآتي :

### DISPLAY MEMORY [TO PRINT]

ويلاحظ هنا وجود اختيار واحد فقط وهو (TO PRINT) . وهو يستخدم عندما يراد طباعة هذه البيانات على الطابعة . ويمكن استخدام حتى ( ٢٥٦ ) متغير ذاكرة بحد أقصى ( ٦٠٠٠ ) حرف . كما يمكن زيادة هذا العدد من خلال ملف المواصفات (Config.sys). وعندما تزيد البيانات المعروضة عن طول الشاشة يتوقف العرض (Pause) وتظهر الرسالة "Press any key to continue".

## ٣٨ - الأمر (DISPLAY STATUS)

يستخدم هذا الأمر في عرض بيانات عن الحالة الحالية للبرنامج . والصورة العامة له كالآتي :

### DISPLAY STATUS [TO PRINT]

وهذا يؤدي الى عرض بيانات عن ملفات قاعدة البيانات المفتوحة ورقم منطقة العمل (Work area) المستخدمة والعلاقات المستخدمة بين الملفات ان وجدت وملفات الفهرس المفتوحة و ... الخ .

## ٣٩ - الأمر (DISPLAY STRUCTURE)

يستخدم هذا الأمر في عرض بيانات عن ملف قاعدة البيانات المفتوح وتركيبه (Structure) الذي يشمل اسم كل حقل ونوعه وعرضه . كما يعرض عدد الحروف (Bytes) التي يتكون منها السجل الواحد . والصورة العامة لهذا الأمر كالآتي :

DISPLAY STRUCTURE [TO PRINT]

وينتج العدد الكلي للحروف (Bytes) في السجل الواحد من مجموع عدد الحروف في جميع الحقول المكونة لهذا السجل بالإضافة إلى حرف آخر يتم حجزه لعلامة المسح التي توضع أمام السجل عندما يراد تجهيزه للمسح .

وعند زيادة عدد سطور السجل عن طول الشاشة يتوقف عرض السطور وتظهر الرسالة "Press any key to continue" . ثم يتم استكمال عرض بيانات السجل عند ضغط المستخدم على أى مفتاح .

## ٤٠ - الأمر (DO)

يستخدم هذا الأمر في تنفيذ برنامج فرعي (Module) أو برنامج خطوات (Procedure) . كما يسمح بادخال المعاملات (Parameters) المطلوبة في البرنامج . والصورة العامة للأمر كالآتي :

DO <filename> [WITH <parameter list>]

ويجب أن يتضمن اسم الملف (filename) رمز وحدة الأقراص أو الفهرس الفرعي الذي يحتوى على ملف البرنامج المطلوب . ولا يتم كتابة الامتداد في اسم الملف . حيث أن البرنامج يضيف الامتداد (.prg) ألياً .

والاختيار (<parameter list> WITH) يستخدم لادخال معاملات (Parameters) الى البرنامج الفرعي الذي يتم تنفيذه . وعندما ينتهي تنفيذ البرنامج الذي تم استدعاؤه بواسطة الأمر (DO) ينتقل التحكم الى البرنامج الذي قام باستدعائه .

## ملاحظة

عدد البرامج التي يتم استدعاؤها بواسطة الأمر (DO) يحسب ضمن عدد الملفات المسموح بفتحها في نفس الوقت . ولكن في حالة استخدام ملف الخطوات أو الإجراءات (Procedure File) فإن فتح هذا الملف يحسب كملف واحد مفتوح بالرغم من أنه يحتوي على العديد من البرامج التي يمكن تنفيذها بواسطة الأمر . (DO) لذلك يفضل استخدام ملف الخطوات أو الإجراءات (Procedure File) على استخدام البرامج المنفصلة (Modules) في أغلب الأحيان .

## مثال ١

لتنفيذ البرنامج (Cadrep) من داخل البرنامج (Cadets) مثلا يتم كتابة الأمر التالي :

```
DO B: Cadrep
```

## مثال ٢

لكتابة البرنامج (Calc) الذي يحسب مساحة المستطيل ، فإن هذا البرنامج يكون كالآتي مثلا :

```
PARAMETERS Length, Width, Area
Area = Length * Width
RETURN
```

ولكي يتم تنفيذ هذا البرنامج مع ادخال الأطوال 8 ، 5 مكان المعاملات (Length) ، (Width) على الترتيب ، يتم كتابة السطور التالية :

```
Area = 0
DO Calc WITH 8,5 , Area
? Area
```

وعند الضغط على مفتاح الادخال يظهر العدد (40) .

والسطر الأول يتم كتابته لحجز مكان في الذاكرة للمتغير (Area) .

## ٤١ - الأمر (DO CASE)

يستخدم هذا الأمر في الانتقال بين مجموعات مختلفة من الأوامر بناءً على شروط معينة يتم ادخالها . وهو يعتبر صورة مكبرة من الأمر (IF) ، حيث أن الأمر (IF) يسمح بالاختيار بين بديلين فقط ، أما هذا الأمر فيسمح بالاختيار بين عدة بدائل . والصورة العامة له كالآتي :

```
DO CASE
  CASE <condition>
    <commands>
  CASE<condition>
    <commands>
  [OTHERWISE]
    <commands>
ENDCASE
```

ويلاحظ أنه يتم التفرع إلى أي حالة (CASE) عندما يصبح الشرط (condition) الخاص بها صحيحاً (True) . والشرط (condition) هو عبارة عن علاقة منطقية (Logical) مثل (A = B) أو (mchoice = 1) . ويبدأ البرنامج باختبار الشرط الأول . فإذا تحقق هذا الشرط يتم تنفيذ الأوامر (Commands) التالية له . وإذا لم يتحقق ينتقل البرنامج إلى الشرط الثاني ... وهكذا .

وعندما ينفذ البرنامج مجموعة من الأوامر التابعة لحالة معينة (CASE) ، فإنه لايقوم باختبار باقي الحالات . وإنما ينتقل إلى الأوامر التالية للأمر (ENDCASE) .

والاختيار (OTHERWISE) اختياري يمكن عدم كتابته . ويستخدم عندما يراد تنفيذ مجموعة من الأوامر في حالة عدم تحقق أي شرط من الشروط السابقة .

## ٤٢ - الأمر (DO WHILE)

يستخدم هذا الأمر في تكوين حلقة تكرارية (Loop) يتم تنفيذها أي عدد من المرات طالما كان الشرط الموجود بعد الأمر صحيحاً (True) . والصورة العامة لهذا الأمر كالآتي :

```
DO WHILE <condition>
    <commands>
ENDDO
```

ويستخدم الأمر (ENDDO) لتحديد نهاية الحلقة . حيث يعود البرنامج الى أول الحلقة ليختبر الشرط مرة ثانية . فإذا تحقق الشرط يتم تنفيذ أوامر الحلقة . وإذا لم يتحقق يتم الانتقال الى الأوامر التي تلي الأمر (ENDDO) . ويمكن استخدام حلقات متداخلة (Nested) . وفي هذه الحالة يجب التأكد من ملاحظة انتهاء كل حلقة داخل الحلقة الخارجية . ولتنظيم هذه العملية يمكن كتابة ملاحظات (Comments) بجوار كل أمر (ENDDO) لتوضيح الحلقة التي يتبعها مع ملاحظة أن أي تعليق (Comment) بجوار الأمر (ENDDO) يتم إهماله بواسطة البرنامج أي أنه يفيد في التوضيح فقط في حالة عرض أوامر البرنامج .

ويمكن استخدام التعويض أو الاحلال بالماكرو (Macro Substitution) ولكن يراعى في هذه الحالة عدم تغيير قيمة هذا الماكرو داخل الحلقة التكرارية .

والبرنامج التالي مثلا يوضح هذه الطريقة .

```
USE Cadets INDEX Name
STORE (UPPER(name) = 'TAREK') TO condition
DO WHILE & condition .AND. .NOT. EOF()
    ? TRIM(name), address
    SKIP
ENDDO
USE
RETURN
```

وهذا البرنامج يقوم أولا بتخزين الشرط الموضح في متغير ذاكرة اسمه (condition) . ثم يقوم باستخدام هذا المتغير في الشرط الموجود بعد الأمر (DO WHILE) .

## ٤٣ - الأمر (EDIT)

يستخدم هذا الأمر في تعديل بيانات سجل معين في ملف قاعدة البيانات . وهو يؤدي نفس العمل الذي يتم عند استخدام برنامج المساعد (Assistant) عن طريق الدخول الى قائمة التحديث (Update) واختيار (EDIT) .



والصورة العامة لهذا الأمر كالآتي :

```
EDIT[<scope>][FIELDS <list>][WHILE <condition>]
[FOR <condition>]
```

وبلاحظ أن جميع الاختيارات اختيارية أى يمكن عدم استخدامها . وفى حالة كتابة الأمر (EDIT) دون كتابة أى شيء بعده تظهر البيانات الخاصة بالسجل الذى يقف عنده مؤشر السجلات (Record Pointer) . وفى هذه الحالة يمكن تعديل البيانات عن طريق تحريك مؤشر التصحيح الى الحقل المطلوب تعديله وكتابة البيانات الجديدة . ويتم الانتقال من سجل الى آخر باستخدام مفتاحى (PgUp) ، (PgDn) . كما يتم تخزين التعديلات التى تم ادخالها بالضغط على مفتاحى (Ctrl-End) .

ولتعديل حقل الملاحظات (memo field) يتم وضع مؤشر التصحيح على هذا الحقل . ثم بالضغط على مفتاحى (Ctrl-PgDn) يتم فتح حقل الملاحظات حيث يتم تعديل الملاحظات أو اضافة ملاحظات جديدة . كما يتم تخزين هذه الملاحظات بالضغط على مفتاحى (Ctrl-PgUp) أو مفتاحى (Ctrl-End) .

#### ٤٤ - الأمر (EJECT)

يستخدم هذا الأمر فى ارسال شفرة نقل الصفحة (Form Feed) الى الطابعة ، وهى عبارة عن شفرة الآسكى (ASCII 12) . ويراعى فى بداية الطباعة أن تبدأ الورقة من أولها .

والصورة العامة للأمر كالآتي :

```
EJECT
```

وهذا الأمر يعيد عداد السطور (Prow()) وعداد الأعمدة (Pcol()) الى الصفر .

#### ٤٥ - الأمر (ERASE)

يستخدم هذا الأمر فى مسح ملف من القرص أو من الفهرس الفرعى . والصورة العامة له كالآتي :

```
ERASE <filename>/?
```

واسم الملف يجب أن يتضمن الامتداد . كما يجب كتابة وحدة الأقراص أو الفهرس الفرعي في حالة عدم وجود الملف المطلوب مسحه على وحدة الأقراص الحالية (Current Drive) . وهذا الأمر يقوم بتنفيذ نفس العمل الذي يتم عند استخدام برنامج المساعد (Assistant) والدخول في قائمة الأدوات (Tools) ، ثم اختيار (ERASE) . كبا أن استخدام هذا الأمر يماثل تماما استخدام الأمر (DELETE FILE) .

ويجب ملاحظة أن هذا الأمر لايسمح باستخدام الحروف الشاملة لمسح عدد من الملفات مثل أوامر نظام التشغيل (MS-DOS) .

## ٤٦ - الأمر (EXIT)

يستخدم هذا الأمر في الخروج من الحلقة التكرارية التي تتكون نتيجة لاستخدام الأمر (DO WHILE) . ويؤدي هذا الأمر الى الانتقال الى الأوامر التي تلى الأمر (ENDDO) .

والصورة العامة له كالآتي :

EXIT

## ٤٧ - الأمر (FIND)

يستخدم هذا الأمر للبحث في ملف قاعدة البيانات المفهرس (Indexed) عن أول سجل يماثل الحقل الفهرسي فيه سلسلة حرفية أو عدد معين يتم ادخاله . وهو يوفر سرعة بحث عالية جدا .

والصورة العامة له كالآتي :

FIND <character string> /<n>

وإذا لم تكن السلسلة الحرفية أو العدد الذي يجري البحث عنه موجودا في قاعدة البيانات تظهر العبارة "No found" على الشاشة ، وفي هذه الحالة يصبح مؤشر السجلات (Record Pointer) عند آخر الملف ، أى أن الداله (EOF()) تصبح صحيحة (True) . وعادة لاتوضع السلسلة الحرفية التي يتم البحث عنها بين علامات تنصيص (Quotation Marks) . ولكن إذا كانت السلسلة الحرفية تحتوى على مسافات خالية في أولها ، فيجب في هذه الحالة وضعها بين علامات تنصيص . ويجب ملاحظة أن المقارنة تصبح صحيحة إذا كانت الحروف الأولى في

السلسلة الحرفية المطلوبة مطابقة للحروف الأولى من حقل المفتاح (Index Key) ،  
بصرف النظر عن باقي الحروف .

وعند البحث عن متغير ذاكرة حرفي يجب إضافة الدالة (&) قبل اسم المتغير .  
وإذا كان هذا المتغير يحتوي على مسافات في بدايته يتم وضع الاسم مع الدالة (&) بين علامات تنصيص (Quotation Marks) ، كالآتي مثلا "&name" .

والأمر (FIND) يماثل الأمر (SEEK) ، ولكن الأمر (SEEK) أشمل منه .  
حيث يمكن استخدام العلاقات الحسابية معه كما سبق الإيضاح في الكتاب الأول .

## مثال

لنبحث عن أي اسم يبدأ بالحرف (M) في ملف الطلبة (Cadets) يتم كتابة  
السطور التالية :

```
. USE Cadets INDEX Name
. FIND M
. ? name
Mohamed
```

يلاحظ في هذه الحالة ظهور الاسم (Mohamed) عند السؤال عن الاسم الموجود  
في الحقل الفهرسي (name) باستخدام الأمر (?) . ويلاحظ أن المقارنة قد تمت  
بين الحرف المطلوب والحرف الأول في حقل الاسم فقط . ولكن في حالة استخدام  
الأمر (SET EXACT ON) ، يصبح الأمر مختلفا . فمثلا عند كتابة الأوامر السابقة  
بالصورة الآتية :

```
. USE Cadets INDEX Name
. SET EXACT ON
. FIND M
No found
```

يلاحظ في هذه الحالة ظهور العبارة "No found" . وذلك لعدم وجود الاسم  
المطابق تماما للحرف (M) ، أي الذي يتكون من الحرف (M) فقط .

## ٤٨ - الأمر (GO/GOTO)

يستخدم هذا الأمر في توجيه مؤشر السجلات (Record Pointer) إلى  
سجل معين . وهناك صورتان لهذا الأمر :

الصورة الأولى تكون كالآتي :

[GO/GOTO] <N>

حيث (N) هو رقم السجل المراد توجيه المؤشر إليه . ويلاحظ أن الأمر (GO/GOTO) اختياري في هذه الحالة . أي يكفي كتابة الرقم فقط دون كتابة الأمر كالآتي مثلا :

- USE Cadets
- 8
- ? RECNO()
- 8

يلاحظ هنا عند السؤال عن رقم السجل الذي يقف عنده المؤشر باستخدام الدالة (RECNO()) ظهور الرقم (8) .

كما يمكن استخدام متغيرات الذاكرة مع الأمر (GO/GOTO) . وذلك كالآتي مثلا :

- STORE 5 TO mnum
- GO mnum
- ? RECNO()
- 5

والصورة الثانية تكون كالآتي :

GO/GOTO BOTTOM/TOP

ويلاحظ هنا أن الأمر (GO/GOTO) ليس اختياريًا كالحالة السابقة . أي أن وجوده ضروري في هذه الحالة .

مثال

- GO TOP
- ? RECNO()
- 1

ويلاحظ ظهور الرقم (1) عند السؤال عن رقم السجل .

## ٤٩ - الأمر (HELP)

يستخدم هذا الأمر في عرض شاشات المساعدة التي يتم من خلالها شرح أوامر البرنامج . والصورة العامة للأمر كالآتي :

HELP keyword

حيث keyword هو أي أمر أو دالة من برنامج (DBase III+) . ويمكن الضغط على مفتاح (F1) الذي يؤدي نفس العمل مثل كتابة الأمر (HELP) . ويؤدي هذا إلى ظهور القائمة الرئيسية لشاشات المساعدة التي يمكن من خلالها الوصول إلى الأمر المطلوب ومتابعة الشرح الخاص به .

## ٥٠ - الأمر (IF)

يستخدم هذا الأمر في تنفيذ مجموعة من الأوامر في حالة تحقق شرط معين . والصورة العامة له كالآتي :

```
IF <condition>
  <commands>
[ELSE]
  <commands>
ENDIF
```

ويقوم البرنامج باختبار الشرط (condition) بعد (IF) فإذا تحقق يتم تنفيذ الأوامر (Commands) التالية للأمر (ELSE) ، ويلاحظ أن الأمر (ELSE) اختياري هنا . أي يمكن عدم كتابته ، وفي هذه الحالة يتم الانتقال إلى الأوامر التي تلي الأمر (ENDIF) في حالة عدم تحقق الشرط .

ويمكن استخدام مجموعات متداخلة من الأوامر (IF) . وفي هذه الحالة يجب ملاحظة وجود الأمر (ENDIF) الخاص بكل مجموعة داخل المجموعة الخارجية . ويمكن ملاحظة ذلك عن طريق كتابة تعليق (Comment) بعد الأمر (ENDIF) . حيث أن هذا التعليق لا يؤثر في تنفيذ البرنامج .

## ٥١ - الأمر (INDEX)

يستخدم هذا الأمر في إنشاء ملف فهرسي يحتوي على حقلي فقط ، أحدهما يحتوي على أرقام السجلات الموجودة في ملف قاعدة البيانات ، والآخر يحتوي على

محتويات الحقل الفهرسي (Index Key) الخاصة بكل سجل . والصورة العامة لهذا الأمر كالتالي :

INDEX ON <index key> TO <filename>[UNIQUE]

حيث :

(Index Key) هو المفتاح الذي يتم الترتيب بناء عليه ، وهو قد يكون حقلا معيناً أو علاقة تشمل عدة حقول . ويجب ملاحظة أن حقول الملاحظات (memo fields) لا تستخدم في هذا المفتاح وكذلك الحقول المنطقية (Logical fields) .  
و (Filename) هو اسم الملف الذي يتم انشاؤه وهو يشمل رمز وحدة الأقراص المستخدمة إذا كان الملف سيتم تخزينه في وحدة أقراص غير الوحدة الحالية (current drive) . كما أن الاسم يضاف إليه الامتداد (.NDX) ألياً .

والاختيار (UNIQUE) هنا اختياري وهو يعني أنه لا توجد داخل الفهرس سوى قيمة واحدة لكل سجل . ويعني ذلك أنه إذا كان هناك أكثر من سجل يحتوى على نفس القيمة للحقل الفهرسي يتم ادخال أول سجل يحتوى على هذه القيمة في ملف الفهرس .

ويجب ملاحظة أن الفهرسة تكون دائماً تصاعدياً (Ascendingly) . كما أنها لا تغير الترتيب الفعلي للسجلات . ولكنها تقوم بترتيب السجلات عند عرضها أو التعامل معها في الذاكرة المؤقتة . أي أنها تعطي صورة مرتبة للسجلات مع الاحتفاظ بالأصل في الملف دون أي تغيير .

ويمكن إضافة مجموعة من الحقول في الحقل الفهرسي . وعند إضافة أنواع مختلفة من الحقول مثل الحقول العددية والحقول التاريخية ، يتم تحويل كل نوع من هذه الأنواع إلى حقول حرفية . حيث يتم تحويل الحقول العددية عن طريق الدالة (STR) ، كما يتم تحويل الحقول التاريخية عن طريق الدالة (DTOC) . وأقصى طول للحقل الفهرسي هو (١٠٠) حرف .

فمثلاً إذا أريد إنشاء ملف فهرسي يحتوى على التاريخ والاسم يتم كتابة السطر التالي :

INDEX ON STR(YEAR(<date>),4)+STR(MONTH(<date>),2)+ ;  
STR(DAY(<date>),2 ) + Name TO <filename>

وفي هذه الحالة يتم ترتيب الملف بناءً على التاريخ أولاً ، ثم يتم ترتيب الأسماء داخل كل تاريخ .

## ٥٢ - الأمر (INPUT)

يستخدم هذا الأمر في توجيه المستخدم إلى إدخال قيمة معينة وتخزين هذه القيمة في متغير ذاكرة . والصورة العامة له كالآتي :

INPUT [<message>] TO <memvar>

حيث :

(message) هي رسالة حرفية تظهر للمستخدم لتوجيهه إلى إدخال المتغير المطلوب . وهي تكون محصورة بين علامات تنصيص فردية 'Single quotes' أو مزدوجة "double quotes" أو أقواس مربعة (Brackets) . وقد تستخدم متغيرات الذاكرة مكان هذه الرسالة ، وفي هذه الحالة لا توضع بين علامات تنصيص (Quotes) .

وعند إدخال المستخدم للقيمة المطلوبة يتم تخزين هذه القيمة في متغير الذاكرة (memvar) . ويتحدد نسوع هذا المتغير حسب نوع الدخالات التي يدخلها المستخدم . فمثلاً عندما يقوم بإدخال عدد ، يتم إنشاء متغير عددي لتخزين هذا العدد . ويجب ملاحظة أن القيم الحرفية يتم إدخالها بين علامات تنصيص (Quotes) .

وهذا الأمر يشبه الأمر (ACCEPT) ولكن الأمر (ACCEPT) يتعامل مع الدخالات الحرفية فقط .

## مثال

لتوجيه المستخدم إلى إدخال اسمه وتخزين هذا الاسم في متغير الذاكرة (mname) يتم كتابة السطر التالي :

INPUT "What is your name" TO mname

## ٥٣ - الأمر (INSERT)

يستخدم هذا الأمر في اضافة سجل جديد الى ملف قاعدة البيانات عند المكان الذي يقف عنده مؤشر السجلات (Record Pointer) والصورة العامة له كالاتي :

INSERT [BLANK][BEFORE]

ويلاحظ هنا أنه يمكن استخدام الأمر (INSERT) بدون أي اختيارات أخرى . وفي هذه الحالة تظهر الحقول الخاصة بسجل جديد يتم ادخال بياناته . وعند الانتهاء من ادخال البيانات يتم تخزين هذه البيانات بالضغط على مفتاحي . (Ctrl-End) وفي هذه الحالة يتم تخزين هذا السجل بعد السجل الحالي (Current Record) الذي يقف عنده المؤشر . وعند استخدام هذا الأمر مع الملف الفهرس (Indexed File) فانه يعمل مثل الأمر (APPEND) تماما . حيث يضيف سجلا في آخر الملف . ثم يضيف سجلات أخرى واحدا تلو الآخر .

والاختيار (BEFORE) يفتح سجلا قبل السجل الحالي . فمثلا اذا كان السجل الحالي هو السجل رقم ٨ فان الأمر (Insert Before) يفتح سجلا جديدا يكون رقمه ( ٨ ) في حين يصبح السجل السابق رقم ( ٩ ) .

والاختيار (BLANK) يؤدي الى اضافة سجل جديد خال مكان السجل الحالي ، بحيث يمكن اضافة البيانات اليه فيما بعد .

## ٥٤ - الأمر (JOIN)

يستخدم هذا الأمر في انشاء ملف جديد عن طريق دمج حقول وسجلات من ملفين مفتوحين . وهو يؤدي عمل الأمر (UPDATE) ولكنه يختلف عنه في أنه ينشئ ملفا جديدا . أما الأمر (UPDATE) فانه يعدل ملفا بناء على ملف آخر .

والصورة العامة لهذا الأمر كالاتي :

JOIN WITH <alias> TO <filename>  
FOR <condition>[FIELDS <fieldlist>]

حيث (filename) هو الملف الجديد الذي يتم انشاؤه ويجب أن يتضمن رمز وحدة الأقراص اذا كان يراد تخزينه على وحدة أقراص غير وحدة الأقراص الحالية (Current Drive) .



و (alias) هو المرادف الذي يشمل اسم الملف المراد دمجها ومنطقة العمل الخاصة بهذا الملف . ويرامى أن يكون الملف المراد دمجها مفتوحاً وأن تكون منطقة العمل (Work Area) الخاصة به قد تم اختيارها (Selected) كمنطقة العمل العاملة (Active) .

والاختيار (<field list> FIELDS) يستخدم لتحديد أسماء الحقول المراد دمجها من الملفين . وفي حالة عدم تحديد هذه الحقول يتم دمج جميع الحقول في الملف الجديد .

### ملاحظة

عندما يراد تحديد حقول من منطقة عمل غير مختارة (Unselected Work Area) يستخدم التعبير field --> (Alias name) حيث (Alias) هو الاسم المرادف لمنطقة العمل غير المختارة .

## ٥٥ - الأمر (LABEL)

يؤدي هذا الأمر إلى استخدام ملف العناوين المختصرة (Label) الذي سبق إنشاؤه . والصورة العامة له كالآتي :

```
LABEL FORM <filename>/?[<scope>][SAMPLE]
[WHILE <condition>][FOR <condition>][TO PRINT]
[TO FILE <filename>]
```

حيث (filename) هو اسم ملف العناوين المختصرة المراد استخدامه ويجب أن يشمل رمز وحدة الأقراص إذا كانت غير وحدة الأقراص المستخدمة (Current drive) .

وتستخدم علامة الاستفهام (?) عندما يراد عرض أسماء ملفات العناوين الموجودة على القرص حتى يستطيع المستخدم اختيار الملف المطلوب استخدامه . و (Scope) هو الذي المطلوب البحث خلاله عن السجلات المطلوب عرض بياناتها في التقرير .

والاختيار (SAMPLE) يستخدم لاختبار ضبط التقرير على الطباعة . ويمكن تكرار عملية الاختبار عدة مرات بكتابة (Y) عند ظهور السؤال التالي :

Do you want more samples?

والاختيار (TO FILE) يستخدم عندما يراد تخزين التقرير في ملف نصوص (Text File) حتى يمكن التعامل معه بواسطة برامج أخرى .  
والاختيارات (WHILE) ، (FOR) تستخدم لتحديد السجلات المطلوب عرض بياناتها في التقرير .

## ٥٦ - الأمر (LIST)

يستخدم هذا الأمر في عرض بيانات ملف قاعدة البيانات . والصورة العامة له كالآتي :

```
LIST [OFF][<scope>][<Field list>][WHILE <condition>]
[FOR <condition>][TO PRINT]
```

ويلاحظ هنا أن جميع الاختيارات مع الأمر اختيارية . أي يمكن كتابته دون كتابة أي شيء بعده . وفي هذه الحالة يتم عرض بيانات جميع سجلات قاعدة البيانات وهو في هذا يختلف عن الأمر (DISPLAY) الذي يعرض بيانات السجل الحالي فقط . كما أن الأمر (LIST) لا يوقف الشاشة عند امتلائها بالبيانات ، بل يتم الانتقال إلى الشاشات التالية فوراً . وعندما يراد إيقاف الشاشة (Pause) يتم الضغط على مفتاحي (Ctrl-S) .

والاختيار (OFF) يستخدم عندما يراد عرض البيانات بدون أرقام السجلات .

والاختيار (scope) يستخدم لتحديد المدى الذي يتم البحث خلاله عن السجلات المراد عرضها .

والاختيار (Field List) يستخدم لتحديد أسماء الحقول المراد عرض بياناتها .

والاختيارات (WHILE) ، (FOR) تستخدم لتحديد السجلات المراد عرض بياناتها .

والاختيار (TO PRINT) يستخدم عندما يراد طباعة البيانات .

## ٥٧ - الأمر (LIST HISTORY)

يستخدم هذا الأمر في عرض آخر عشرين أمرا سبق تنفيذها وهو يماثل الأمر (DISPLAY HISTORY) ولكنه لا يؤدي الى توقف الشاشة عندما تزيد الأوامر عن طول الشاشة . والصورة العامة له كالآتي :

LIST HISTORY [LAST <N>] [TO PRINT]

والاختيار (LAST <N>) يستخدم لتحديد عدد الأوامر المطلوب عرضها .

والاختيار (TO PRINT) يستخدم لطباعة الأوامر المعروضة .

## ٥٨ - الأمر (LIST MEMORY)

يستخدم هذا الأمر في عرض حالة متغيرات الذاكرة (Memory Variables) الموجودة في الذاكرة المؤقتة (RAM) في أي وقت ، حيث يعرض اسم كل متغير ونوعه وحجمه ، كما يعرض أيضا بيانات عن حجم الذاكرة المستخدمة وحجم الذاكرة المتاحة . والصورة العامة لهذا الأمر كالآتي :

LIST MEMORY [TO PRINT]

وهو يماثل الأمر (DISPLAY MEMORY) تماما ويختلف عنه فقط في أنه لا يوقف عرض البيانات عندما تزيد عن طول الشاشة .

## ٥٩ - الأمر (LIST STATUS)

يستخدم هذا الأمر في عرض بيانات عن الحالة الحالية للبرنامج . والصورة العامة له كالآتي :

LIST STATUS [TO PRINT]

وهو يؤدي الى عرض بيانات عن ملفات قاعدة البيانات المفتوحة ورقم منطقة العمل (Work Area) المستخدمة والعلاقات المستخدمة بين الملفات ان وجدت وملفات الفهرس المفتوحة و ... الخ .

وهذا الأمر يماثل الأمر (DISPLAY STATUS) تماما ولكنه لا يتوقف عند امتلاء الشاشة بالبيانات .

## ٦٠ - الأمر (LIST STRUCTURE)

يستخدم هذا الأمر في عرض بيانات عن ملف قاعدة البيانات المفتوح وتركيبه (Structure) الذي يشمل اسم كل حقل ونوعه وعرضه . كما يمرض عدد الحروف (Bytes) التي يتكون منها السجل الواحد . والصورة العامة لهذا الأمر كالآتي :

LIST STRUCTURE [TO PRINT]

وهو يماثل الأمر (DISPLAY STRUCTURE) تماما . ولكنه لا يتوقف عند امتلاء الشاشة بالبيانات .

## ٦١ - الأمر (LOAD)

يستخدم هذا الأمر في كتابة برامج فرعية بلغة التجميع (Assembly) وتشغيل هذه البرامج من خلال البرنامج الذي يتم تصميمه بواسطة (DBase III+) . ويجب أن تكون هذه البرامج على هيئة الشفرة الثنائية (Binary Code) . والصورة العامة لهذا الأمر كالآتي :

LOAD <filename>

وهذا يؤدي الى تحميل الملف (Filename) في الذاكرة المؤقتة حتى يتم استدعاؤه في البرنامج بواسطة الأمر (CALL) . ويسمح برنامج (DBase III+) بتحميل حتى خمسة برامج من هذا النوع في كل مرة . وكل برنامج بحسب أقصى ٢٢ ك بايت بشرط ان تكون بالشفرة الثنائية (Binary) .

## ٦٢ - الأمر (LOCATE)

يستخدم هذا الأمر للبحث في ملف قاعدة البيانات عن السجل الذي يحقق شرطا أو شروطا معينة . والصورة العامة له كالآتي :

LOCATE [<scope>] [WHILE <condition>] [FOR <condition>]

والاختيار (scope) يستخدم لتحديد مدى معين من السجلات للبحث خلاله .  
فمثلا عند كتابة (NEXT <N>) فإن البحث يتم خلال العدد (N) من السجلات التي  
تلي السجل الحالي (Current Record) .

والاختيارات (WHILE) ، (FOR) تستخدم لتحديد السجلات المطلوب البحث  
عنها . ويجب ملاحظة أن هذا الأمر يؤدي إلى توجيه المؤشر إلى أول سجل يحقق  
الشروط . فإذا أريد تحريكه إلى باقي السجلات يستخدم الأمر (CONTINUE) .

### ٦٣ - الأمر (LOOP)

يستخدم هذا الأمر في الانتقال إلى بداية الحلقة التكرارية دون تنفيذ باقي أوامر  
الحلقة . والصورة العامة له كالآتي :

LOOP

ويستخدم في معظم الأحيان في البرامج الفرعية (Modules) الخاصة بقوائم  
الاختيارات (Menus) ، فمثلا يمكن كتابة الأوامر التالية في قاعدة بيانات الطلبة  
: (Cadets)

```
USE Cadets
  mchoice = " "
  CLEAR
  @ 5,5 SAY "A-Add records"
  @ 7,5 SAY "E-Edit records"
  @ 9,5 SAY "R-Return to dot prompt"
  READ
  IF .NOT. mchoice $ "AaEeRr"
    LOOP
  ENDIF
DO CASE
  CASE mchoice $ "Aa"
    APPEND
  CASE mchoice $ "Ee"
    EDIT
  CASE mchoice $ "Rr"
    EXIT
ENDCASE
ENDDO
CLOSE DATABASES
```

ويتم اختبار المتغير (mchoice) بعد الأمر (IF) ، فإذا لم يكن الحرف المخزن به ضمن الحروف (AaEeRr) يتم تنفيذ الأمر التالي وهو (LOOP) . وفي هذه الحالة يتم الانتقال إلى أول الحلقة التكرارية مرة ثانية لاختبار الحرف الجديد الذي يدخله المستخدم في المتغير (mchoice) . والدالة (\$) تعني هنا البحث عن الحرف الموجود في المتغير (mchoice) خلال السلسلة الحرفية (AaEeRr) . وفي هذه السلسلة تم وضع الحروف المثلة لاختيارات القائمة سواء كانت صغيرة (small) أو كبيرة (Capital) .

## ٦٤ - الأمر (MODIFY COMMAND)

يستخدم هذا الأمر في إنشاء وتعديل ملفات الأوامر (Command files) التي تستخدم في كتابة البرامج الخاصة ببرنامج (DBase III+) . وهذا الأمر يؤدي إلى تشغيل برنامج تنسيق النصوص (Text editor) الخاص ببرنامج (DBase III+) والذي يمكن استخدامه في كتابة أي نص (Text) .

والصورة العامة للأمر كالآتي :

MODIFY COMMAND <filename>

ويمكن أن يحتوي اسم الملف (filename) على رمز وحدة الأقراص المراد تخزين الملف عليها إذا لم تكن هي وحدة الأقراص الحالية (Current Drive) . وهذا الأمر يضيف الامتداد (.prg) إلى اسم الملف آلياً . أما إذا أريد إضافة أي امتداد آخر، فيجب في هذه الحالة إضافة الامتداد إلى اسم الملف المراد إنشاؤه .

وعند كتابة هذا الأمر فإن البرنامج يبحث عن ملف بالاسم (Filename) فإذا وجدته فإنه يستدعيه ويعرض الملف لتصحيحه . وإذا لم يجده فإنه ينشئ ملفاً جديداً بهذا الاسم .

وهذا الأمر يسمح بإنشاء ملف حتى ٥٠٠٠ حرف (Byte) . أما إذا استخدم برنامج تنسيق نصوص خارجي من خلال ملف المواصفات (Configuration File) فإنه يمكن في هذه الحالة زيادة عدد الحروف عن ذلك .

وعند كتابة أمر يزيد عن طول السطر يمكن كتابة الفاصلة المنقوطة (;) في نهاية السطر ثم استكمال كتابة الأمر في السطر التالي .

وعندما يراد طباعة محتويات الملف يستخدم الأمر  
(TYPE <filename> TO PRINT) من مشيرة النقطة (Dot Prompt) .

#### ٦٥ - الأمر (MODIFY LABEL)

ارجع الى (CREATE/MODIFY LABEL)

#### ٦٦ - الأمر (MODIFY QUERY)

ارجع الى (CREATE/MODIFY QUERY)

#### ٦٧ - الأمر (MODIFY REPORT)

ارجع الى (CREATE/MODIFY REPORT)

#### ٦٨ - الأمر (MODIFY SCREEN)

ارجع الى (CREATE/MODIFY SCREEN)

#### ٦٩ - الأمر (MODIFY STRUCTURE)

ويستخدم هذا الأمر عندما يراد تعديل تركيب ملف قاعدة البيانات والصورة العامة له كالآتي :

MODIFY STRUCTURE <Filename>

وهذا الأمر ينشئ دائما ملفا احتياطيا بالامتداد (.bak) لكل ملف يتم تعديله . وعند الانتهاء من ادخال التعديلات ، يقوم باضافة السجلات (Append) الى الملف الأصلي . وإذا تم تعديل أي حقول ملاحظات ، يتم انشاء ملف احتياطي لملف الملاحظات بالامتداد (&&&) .

## ٧٠ - الأمر (NOTE/\*)

يستخدم الأمر (NOTE) أو النجمة (\*) لنفس الهدف وهو كتابة ملاحظات لتوضيح وظائف أوامر البرنامج المختلفة . ويتم كتابة هذا الأمر في بداية السطر المطلوب كتابة ملاحظات فيه . والصورة العامة لهذا الأمر كالآتي :

NOTE/\* <text>

وعندما يراد كتابة عدة سطور ملاحظات متتالية ، يمكن كتابة فاصلة منقوطة ( ; ) في نهاية كل سطر . ويمكن كتابة الملاحظات في أى مكان في السطر باستخدام الحرفين (&&) .

### مثال

هذا المثال يوضح استخدام الأمر (NOTE) في برنامج .

```
NOTE This is a loop
STORE 1 TO X
DO WHILE X < 100
    STORE X + 1 TO X
ENDDO
```

كما يمكن استخدام الحرفين (&&) كالآتي مثلا :

```
mname = SPACE (35) && initiate a memory variable
```

## ٧١ - الأمر (ON)

يستخدم هذا الأمر في تنفيذ أمر أو برنامج فرعي بناء على تحقيق شرط معين . والصورة العامة له كالآتي :

ON ERROR/ESCAPE/KEY <command>

وهناك ثلاثة صور للأمر كما هو واضح وهي كالآتي :

ON ERROR <command> وهو يؤدي إلى تنفيذ أمر معين أو مجموعة من الأوامر في حالة ظهور خطأ (Error) في البرنامج .



ON ESCAPE <command> وهو يؤدي إلى تنفيذ أمر معين أو مجموعة من الأوامر في حالة ضغط المستخدم على مفتاح الهروب (ESC) .

ON KEY <command> وهو يؤدي إلى تنفيذ أمر معين أو مجموعة من الأوامر في حالة ضغط المستخدم على أي مفتاح .

ويظل هذا الأمر مؤثرا في البرنامج حتى إنهائه بواسطة الأمر (ON ERROR/ESCAPE/KEY) ، وذلك دون كتابة أي شيء بعده .

ويجب ملاحظة أن الأمر (ON ESCAPE) لايعمل مع الأمر (SET ESCAPE OFF) . كما يجب ملاحظة أن الأمر (ON ESCAPE) له أولوية على الأمر (ON KEY) . وهذا يعني أنه عند كتابة الأمرين معا في نفس البرنامج فإن الضغط على مفتاح الهروب يؤدي إلى تنفيذ الأمر أو الأوامر التي تلي الأمر (ON ESCAPE) وليس الأوامر التي تلي الأمر (ON KEY) .

والأمر (ON ERROR) يتعامل فقط مع الأخطاء المتعلقة بقواعد أوامر (DBase III+) ، وهي التي تسمى أخطاء القواعد (Syntax Errors) .

## مثال

يمكن تشغيل البرنامج (error1) عند ظهور أي خطأ في البرنامج ، وذلك عن طريق كتابة السطر التالي :

ON ERROR DO error1

كما يمكن كتابة الأمر التالي :

ON ESCAPE EXIT

وذلك للخروج من حلقة تكرارية مثلا في حالة ضغط المستخدم على مفتاح الهروب (ESC) . كما يمكن إيقاف تأثير مفتاح الهروب عن طريق كتابة الأمر (ON ESCAPE) دون كتابة أي شيء بعده .

## ٧٢ - الأمر (PACK)

يستخدم هذا الأمر في مسح السجلات التي سبق وضع علامات عليها تمهيدا لمسحها من الملف . والصورة العامة له كالآتي :

PACK

## ٧٣ - الأمر (PARAMETERS)

ويستخدم هذا الأمر في تخصيص معاملات في البرنامج الفرعي يتم من خلالها استقبال القيم التي يتم ادخالها عندها استدعاء هذا البرنامج الفرعي . وعند استدعاء هذا البرنامج الفرعي يستخدم الأمر (DO <filename> WITH <parameter list>) حيث يتم ادخال القيم المقابلة للمتغيرات التي سبق تخصيصها . والصورة العامة لهذا الأمر كالآتي :

PARAMETERS <parameter list>

ويكتب هذا الأمر في بداية البرنامج الفرعي سواء كان برنامج خطوات (Procedure) أو أي برنامج فرعي آخر .

وقائمة المعاملات (Parameter list) هي المتغيرات التي تستقبل المعاملات التي يتم امرها مع الأمر (DO) . ويجب أن يكون عدد المعاملات في أمر الاستقبال (PARAMETERS) وأمر الأوامر (DO) واحدا .

وإذا كان المعامل الذي يتم ادخاله أو امره (Passing) متغير ذاكرة ، فإن قيمة هذا المتغير قد تتغير بناء على تشغيل البرنامج الفرعي ، ثم يتم نقل هذه القيمة إلى البرنامج الذي قام باستدعاء البرنامج الفرعي .

## مثال

الأوامر التالية تمثل برنامج خطوات أو إجراءات (Procedure) اسمه (Calc) :

```
PARAMETERS length , width , area
area = length * width
RETURN
```

ولتنفيذ هذا البرنامج مع امرار القيمة (7) للطول (Length) . والقيمة (9) للعرض (width) يتم كتابة الأمر التالي :

```
DO Calc WITH 7, 9, area
```

وللإزالة عن القيمة المخزنة في المتغير (area) يتم كتابة الأمر التالي :

? area

وفي هذه الحالة يظهر العدد (63) الذى يمثل نتيجة المعادلة الموجودة فى البرنامج (Calc).

## ٧٤ - الأمر (PRIVATE)

يستخدم هذا الأمر عندما يراد استخدام متغيرات ذاكرة فى برنامج فرعى بحيث لا تتأثر بأى متغيرات أخرى تحمل نفس أسماء هذه المتغيرات وتكون موجودة فى البرنامج الرئيسى أو فى أى برامج أخرى غير البرامج التابعة لهذا البرنامج الفرعى . كما يستخدم هذا الأمر أيضا لتغيير حالة متغير الذاكرة من متغير عام (Public) الى متغير خاص (Private). وهناك صورتان للأمر كالآتى :

```
PRIVATE <memvar list>
```

```
PRIVATE ALL [LIKE/EXCEPT <skeleton>]
```

ويستخدم الاختيار (LIKE) عندما يراد استخدام الحروف الشاملة فى تغيير عدد من المتغيرات العامة (Public) الى خاصة (Private) . كما يستخدم الاختيار (EXCEPT) عندما يراد تحويل كل المتغيرات العامة الى متغيرات خاصة ما عدا بعض المتغيرات التى يتم تحديدها عن طريق استخدام الحروف الشاملة (Global Characters).

فمثلا لتحويل كل المتغيرات التى تبدأ بالحرف (M) الى متغيرات خاصة يستخدم الأمر التالى :

```
PRIVATE ALL LIKE M*
```

كما يمكن تحويل كل المتغيرات الى متغيرات خاصة ماعدا المتغيرات التى تبدأ بالحرف (S) كالآتى :

```
PRIVATE ALL EXCEPT S*
```

ويجب ملاحظة أن المتغيرات الخاصة تختفى القيم الموجودة بها بمجرد انتهاء البرنامج ، وذلك عكس للمتغيرات العامة (Public) التى تظل محتفظة بأخر قيمة تم ادخالها فيها .

## ٧٥ - الأمر (PROCEDURE)

يستخدم هذا الأمر لتحديد بداية كل برنامج خطوات او اجراءات

(Procedure) داخل ملف الخطوات (Procedure file) . والصورة العامة له كالآتي :

PROCEDURE <procedure name>

وكل برنامج خطوات أو إجراءات (Procedure) يبدأ بالأمر (PROCEDURE) يليه اسم برنامج الخطوات . واسم برنامج الخطوات يتكون من ٨ حروف . ويمكن أن يحتوي على حروف أو أرقام أو شرطة سفلية (Underscore) . كما يجب أن يبدأ بحرف (Character) .

### مثال

يمكن أن يحتوي ملف الخطوات أو الإجراءات (Proc1) على برامج الخطوات الآتية :

```
PROCEDURE message
    @ 10,0 CLEAR
    @ 15,0 SAY "This is an invalid data"
RETURN
PROCEDURE printer
    @ 10,0 CLEAR
    @ 15,0 SAY "Send reprot to printer? Y/N" GET Pr;
    PICTURE!"
    IF Pr = Y
        SET PRINT ON
    ENDIF
RETURN
```

ولتشغيل برنامج الخطوات (Printer) مثلا ، يتم أولا فتح ملف الخطوات (Proc1) وذلك كالآتي :

```
SET PROCEDURE TO Proc1
DO Printer
```

### ٧٦ - الأمر (PUBLIC)

يستخدم هذا الأمر في تحويل المتغيرات إلى متغيرات عامة يمكن استخدامها في أي برنامج فرعي أو رئيسي . وهذه المتغيرات تختلف عن باقي متغيرات الذاكرة في أن القيم المخزنة بها تظل موجودة حتى بعد انتهاء البرنامج ، وتظل موجودة في الذاكرة

المؤقتة حتى يتم مسحها باستخدام الأمر (RELEASE) أو (CLEAR MEMORY) . والصورة العامة للأمر كالآتي :

PUBLIC <memory variable list>

وعند استخدام هذا الأمر فإن أى برنامج فرعى أو رئيسى يستخدم هذه المتغيرات يمكنه تغيير القيم للوجود فيها . ويجب ملاحظة أن المتغيرات العامة يجب إعلانها عامة (Declaring) قبل إعطائها أى قيمة . فمثلا الأوامر التالية تعتبر خطأ :

```
Answer = "Y"
PUBLIC Answer
```

وذلك لأن المتغير (Answer) قد تم إعطاؤه القيمة "Y" قبل إعلانه عاما .

## ٧٧ - الأمر (QUIT)

يستخدم هذا الأمر فى إغلاق جميع الملفات والخروج من البرنامج والرجوع الى نظام التشغيل . والصورة العامة له كالآتي :

QUIT

ويساعد هذا الأمر على الخروج من البرنامج بطريقة آمنة دون تحطيم أى ملفات ، حيث أن إغلاق جهاز الحاسب دون استخدام هذا الأمر قد يؤدى الى تحطيم الملفات المفتوحة وفقد البيانات المخزنة فيها .

## ٧٨ - الأمر (READ)

يستخدم هذا الأمر فى تخزين البيانات التى يتم ادخالها عن طريق الأمر (@...GET) . وهو يستخدم عادة فى تصميم شاشات الادخال من خلال البرنامج . والصورة العامة له كالآتي :

READ [SAVE]

والاختيار (SAVE) يستخدم للإبقاء على البيانات مخزنة في الـ (GETS) حتى يتم مسحها بواسطة الأمر (CLEAR GETS) ، حيث أن الوضع الطبيعي أن يتم مسح هذه الـ (GETS) بمجرد ادخال البيانات في متغيرات الذاكرة ، فمثلا عند كتابة الأوامر التالية :

```
STORE " " TO mname
@ 10,10 SAY "Enter your name:" GET mname
READ
```

في هذه الحالة يتم تخزين الاسم الذي يدخله المستخدم في مخزن مؤقت (GET) . وعندما يقابل البرنامج الأمر (READ) فإنه ينقل هذا الاسم الى متغير الذاكرة (mname) ، وبالتالي يتم إخلاء المخزن المؤقت (GET) .

أما عند استخدام الأمر (READ SAVE) بدلا من الأمر (READ) في المثال السابق ، فإن المخزن المؤقت (GET) يظل محتفظا بالاسم الذي تم ادخاله ، ولذلك يراعى عند استخدام الأمر (READ SAVE) عدة مرات التأكد من مسح المخازن المؤقتة (GETS) قبل أن يصل عددها الى الحد الأقصى المسموح به في البرنامج الذي يكون (١٢٨) مخزنا في الوضع البدئي (Default) .

ويستخدم الأمر (READ) أيضا عند تصميم شاشات الادخال التي تتضمن أكثر من شاشة ، حيث يتم كتابة الأمر (READ) في نهاية كل شاشة ويصبح في هذه الحالة ناقل للصفحة التالية (Page break) . ويستطيع المستخدم في هذه الحالة الانتقال من أي شاشة الى الشاشة الأخرى باستخدام مفتاحي (PgUp) ، (PgDn) .

## ٧٩ - الأمر (RECALL)

يستخدم هذا الأمر في استعادة السجلات التي سبق وضع علامات عليها تمهيدا لمسحها باستخدام الأمر (PACK) . والصورة العامة له كالآتي :

```
RECALL [<scope>][WHILE <condition>][FOR <condition>]
```

والاختيارات كلها اختيارية في هذه الحالة ، حيث يمكن كتابة الأمر دون كتابة أي شيء بعده ، وفي هذه الحالة يتم استعادة أول سجل فقط من السجلات التي سبق وضع علامات عليها لمسحها . وتستخدم الاختيارات في استعادة السجلات التي تحقق شروطا معينة .

فمثلا لاستعادة أول سجل تم تجهيزه للمسح ، يتم كتابة السطر التالي :

RECALL

وإذا أريد استعادة السجل رقم ( ١٠ ) يتم كتابة السطر التالي :

RECALL RECORD 10

وإذا أريد استعادة جميع السجلات التي تم تجهيزها للمسح ، يتم كتابة السطر التالي :

RECALL ALL

حيث تمثل (ALL) المدى (scope) الذي يشمل كل السجلات التي تم تجهيزها للمسح .

#### ٨٠ - الأمر (REINDEX)

يستخدم هذا الأمر في إعادة انشاء ملف الفهرس المفتوح ، وذلك حتى يتم تحديث هذا الملف في بعض الحالات التي تتطلب ذلك . والصورة العامة له كالآتي :

REINDEX

ويجب مراعاة فتح ملف الفهرس المراد تحديثه قبل كتابة هذا الأمر .

#### ٨١ - الأمر (RELEASE)

يستخدم هذا الأمر في مسح متغيرات الذاكرة (Memory Variables) مما يتيح استخدام الذاكرة المتاحة في ادخال متغيرات أخرى .

والصورة العامة له كالآتي :

RELEASE <memvar list> [ALL][LIKE/EXCEPT <skeleton>]  
[MODULE <module name>]

ويستخدم الاختيار (skeleton) عندما يراد استخدام الحروف الشاملة

(Global) في تحديد المتغيرات المطلوب مسحها أو استثناء بعض المتغيرات من المسح حسب الحاجة .

كما يستخدم الاختيار (LIKE) لتحديد المتغيرات المشتركة في حرف معين أو عدة حروف .

ويستخدم الاختيار (EXCEPT) في استثناء المتغيرات المشتركة في حرف معين أو عدة حروف .

كما يستخدم الاختيار (ALL) في مسح جميع متغيرات الذاكرة .

ويستخدم الاختيار (MODULE) في مسح برنامج مكتوب بلغة التجميع (Assembly) من الذاكرة ، ويساعد هذا على استخدام عدة برامج خارجية تزيد من الحد الأقصى لعدد هذه البرامج وهو خمسة .

### أمثلة

لمسح كل متغيرات الذاكرة التي تبدأ بالحرف (m) يتم كتابة السطر التالي :

```
RELEASE ALL LIKE m *
```

ولمسح جميع متغيرات الذاكرة ما عدا المتغيرات التي تبدأ بحرف (s) يتم كتابة السطر التالي :

```
RELEASE ALL EXCEPT s *
```

كما يمكن مسح متغيرات ذاكرة معينة بكتابة السطر التالي مثلا :

```
RELEASE mname , maddress , mage
```

### ٨٢ - الأمر (RENAME)

يستخدم هذا الأمر في تغيير اسم ملف سبق تخزينه على القرص الى اسم جديد . والصورة العامة لهذا الأمر كالآتي :

```
RENAME <old filename> TO <new filename>
```



ويجب أن يتضمن الاسم القديم والاسم الجديد الامتداد ورمز وحدة الأقراص المخزن عليها الملف إذا كانت غير وحدة الأقراص الحالية (Current drive). وفي حالة تغيير اسم ملف قاعدة بيانات يحتوي على حقول ملاحظات (memo fields) يتم تغيير اسم ملف الملاحظات منفصلاً .

## مثال

لتغيير اسم الملف (Cadets) الى الاسم (Grades) يتم كتابة السطر التالي :

```
RENAME Cadets.dbf TO Grades.dbf
```

## ٨٣ - الأمر (REPLACE)

يستخدم هذا الأمر في استبدال محتويات حقول معينة بقيمة جديدة يتم ادخالها . ويتم الحصول على هذه القيمة عادة من متغيرات ذاكرة (Memory Variables). والصورة العامة لهذا الأمر كالآتي :

```
REPLACE[<scope>]<field1>
WITH <exp>[,<field2> WITH <exp>]
[WHILE <condition>][FOR <condition>]
```

وفي حالة عدم ادخال مدى (scope) أو شروط معينة للبحث فإن الاستبدال يتم على السجل الحالي (Current Record) . كما يتم تحديد الحقول التي يتم استبدالها عن طريق كتابة اسم كل حقول مكان الاختيارات (field1, field2, ...etc) . ويمكن استبدال جميع حقول السجل بقيمة جديدة .

وفي حالة استبدال الحقل الفهرسي (Key Field) بقيمة جديدة فإن ذلك سوف يؤدي الى تحديث الفهرس (Updating) ، وبالتالي يحدث تغيير في ترتيب السجلات ، لذلك يراعى عند استبدال الحقل الفهرسي عدم استخدام المدى (scope) وأوامر البحث مثل (WHILE , FOR) لأن ترتيب السجلات يكون قد تغير .

## مثال

لزيادة سعر الوحدة (Unit Cost) بمقدار ( ١٠ × ) في جميع السجلات ،

يتم كتابة السطر التالي :

REPLACE ALL Unitcost WITH Unitcost\*1.1

## ٨٤ - الأمر (REPORT)

يستخدم هذا الأمر لعرض التقرير الذي سبق تصميمه متضمنا بيانات مجموعة من السجلات يتم تحديدها ، وقد يتم عرض التقرير على الشاشة أو طباعته على الطابعة أو تخزينه كملف آسكي (ASCII file) . والصورة العامة لهذا الأمر كالآتي :

```
REPORT FORM <filename>/?[<scope>] [WHILE <condition>]
[FOR <condition>][PLAIN][HEADING <exp>][NOEJECT]
[TO PRINT/TO FILE <filename>][SUMMARY]
```

ولا يلزم هنا كتابة الامتداد في اسم الملف (filename) حيث أن البرنامج يضيف الامتداد (.fmt) أليا .

ويمكن تلخيص الاختيارات المختلفة كالآتي :

الاختيارات (scope) ، (WHILE) ، (FOR) تستخدم لتحديد مدى وشروط البحث التي يتم عن طريقها تحديد السجلات المطلوب عرض بياناتها .

والاختيار (PLAIN) يؤدي الى عرض تقرير عادي لا يحتوى على أرقام الصفحات أو تاريخ اليوم الحالي .

والاختيار (HEADING) يستخدم في اضافة سلسلة حرفية تمثل عنوانا معيناً للتقرير يكتب في كل صفحة .

والاختيار (NOEJECT) يستخدم لافاء نقل الصفحة (Page Break) ، وهذا يتيح لمخطط البرامج التحكم في مكان الانتقال الى الصفحة التالية وبالتالي يمكنه التحكم في طول الصفحة (Page Length) .

الاختيار (TO PRINT) يستخدم عندما يراد طباعة التقرير .

الاختيار (TO FILE) يستخدم لتخزين التقرير كملف آسكي (ASCII file) ، وهذا يتيح التعامل معه من خلال برامج أخرى مثل برامج الجداول الالكترونية .

الاختيار (SUMMARY) يستخدم في الحصول على تقرير مختصر يحتوي فقط على جميع البيانات العددية الموجودة في الملف ، وفي حالة استخدام الاختيار (Groups) عند انشاء التقرير من خلال برنامج المساعد (Assistant) ، يتم تجميع البيانات الخاصة بكل مجموعة منفصلة . ويراعي في هذه الحالة فهرسة الملف باستخدام الحقل المستخدم في التجميع (Grouping) كحقل فهرسي (Key field) .

## ٨٥ - الأمر (RESTORE)

يستخدم هذا الأمر في استرجاع متغيرات الذاكرة التي سبق تخزينها في ملف ذاكرة (Memory file) . والصورة العامة له كالآتي :

RESTORE FROM <filename> [ADDITIVE]

حيث (filename) هو اسم ملف الذاكرة المطلوب استرجاعه .

والاختيار (ADDITIVE) يستخدم لاضافة المتغيرات الموجودة في الملف الى المتغيرات الموجودة في الذاكرة المؤقتة في هذا الوقت ، حيث أن عدم استخدام هذا الاختيار يؤدي الى مسح جميع المتغيرات الموجودة في الذاكرة لتحل محلها المتغيرات الموجودة في الملف .

ويجب ملاحظة أن جميع المتغيرات التي يتم استرجاعها تصبح خاصة (Private) بصرف النظر عن حالتها داخل الملف . وعند استرجاع الملف من مشيرة النقطة (Dot Prompt) تصبح المتغيرات عامة (Public) .

## ٨٦ - الأمر (RESUME)

يستخدم هذا الأمر لاستكمال تنفيذ البرنامج بعد ايقافه بواسطة الأمر (SUSPEND) . والصورة العامة له كالآتي :

RESUME

ويؤدي هذا الأمر الى استكمال تنفيذ البرنامج من نفس المكان الذي توقف عنده عند استخدام الأمر (SUSPEND) . ويستخدم هذا عادة في اختبار وتصحيح البرنامج (Testing and Debugging) .

## ٨٧ - الأمر (RETURN)

يستخدم هذا الأمر في الرجوع من البرنامج الفرعي إلى البرنامج الذي قام باستدعائه ، أو الرجوع إلى مشيرة النقطة (Dot Prompt) في حالة استدعائه من البرنامج الرئيسي . والصورة العامة له كالآتي :

RETURN [TO MASTER]

والاختيار (TO MASTER) يستخدم للرجوع إلى البرنامج الرئيسي مباشرة من أي برنامج فرعي . وعند الرجوع إلى البرنامج القائم باستدعاء البرنامج الفرعي يتم استكمال تنفيذ الأوامر التي تلي الأمر الذي قام باستدعائه .

واستخدام الأمر (RETURN) في نهاية البرنامج يؤدي إلى إغلاق هذا البرنامج، كما يؤدي إلى مسح متغيرات الذاكرة الخاصة (Private) ولكنه لا يؤثر في المتغيرات العامة (Public) .

## ٨٨ - الأمر (RUN/I)

يستخدم هذا الأمر في تشغيل أمر من أوامر نظام التشغيل (MS-DOS) من خلال

برنامج (DBase III+) . والصورة العامة له كالآتي :

RUN <command>

كما يمكن كتابته كالآتي :

!<command>

حيث يعمل الحرف (!) نفس عمل الأمر (RUN) .

ويمكن من خلال هذا الأمر تشغيل أوامر نظام التشغيل مثل (DIR, RENAME, ERASE) .

ويجب ملاحظة أن هذا الأمر يتطلب ذاكرة مؤقتة أكبر من (٢٥٦) حرفا .

## ٨٩ - الأمر (SAVE)

يستخدم هذا الأمر في تخزين متغيرات الذاكرة الموجودة في الذاكرة المؤقتة في ملف ذاكرة (Memory File) . والصورة العامة له كالآتي :

SAVE TO <filename> [ALL LIKE/EXCEPT <skeleton>]

حيث (filename) هو اسم الملف المطلوب تخزين متغيرات الذاكرة فيه . ويجب أن يحتوي اسم الملف على رمز وحدة الأقراص المراد تخزين الملف فيها إذا كانت غير وحدة الأقراص الحالية (Current Drive) . ولا يتم كتابة الامتداد حيث أن البرنامج يضيف الامتداد (.mem) ألياً .

والاختيار (ALL LIKE) يستخدم عندما يراد استخدام الحروف الشاملة في تحديد مجموعة معينة من متغيرات الذاكرة التي تحتوي أسماؤها على حروف معينة .

والاختيار (ALL EXCEPT) يستخدم عندما يراد استثناء بعض المتغيرات التي تحتوي على حروف معينة من التخزين في الملف .

والاختيار (skeleton) يستخدم لادخال الحروف الشاملة (Global Characters) المطلوب استخدامها .

وهناك حرفان يستخدمان كحروف شاملة وهما الحرف (\*) وهو يحل محل أى عدد من الحروف والحرف (?) ويحل محل حرف واحد فقط .

## أمثلة

عندما يراد تخزين جميع متغيرات الذاكرة الموجودة في الذاكرة المؤقتة في ملف اسمه (File1) مثلاً يتم كتابة السطر التالي :

SAVE TO B: file1

ولتخزين جميع متغيرات الذاكرة الموجودة في الذاكرة المؤقتة والتي تبدأ بالحرف (m) في الملف (File2) الموجود على وحدة الأقراص (B) يتم كتابة السطر التالي :

SAVE ALL LIKE m\* TO B: file2

ويجب ملاحظة أنه عند تخزين متغيرات ذاكرة في ملف سبق انشاؤه تظهر الرسالة التالية :

file2.mem already exists, overwrite it? (Y/N)

وعند كتابة (Y) فإن للتغيرات الجديدة تسمح أى متغيرات أخرى سبق تخزينها في الملف .

ولتخزين جميع المتغيرات التي لا تحتوى على الحرف (A) كحرف ثان في اسم المتغير يتم كتابة السطر التالي :

SAVE ALL EXCEPT ?A\*

حيث يمثل الحرف الشامل (?) الحرف الأول من اسم المتغير ، بينما يمثل الحرف الشامل (\*) الحروف الباقية من اسم متغير الذاكرة .

## ٩٠ - الأمر (SEEK)

يستخدم هذا الأمر في البحث خلال ملف قاعدة البيانات المفهرس (Indexed) عن السجل الذي تطابق محتويات الحقل المفهرسي له قيمة معينة يتم ادخالها . والصورة العامة له كالآتي :

SEEK <expression>

والتعبير (expression) قد يكون قيمة عددية أو حرفية أو تاريخية أو علاقة حسابية . وإذا كان قيمة حرفية (string) فيجب كتابتها بين علامات تنصيص (Quotation) . وهو في هذا يختلف عن الأمر الآخر (FIND) الذي لا يحتاج إلى وضع القيمة الحرفية المطلوب البحث عنها بين علامات تنصيص .

ويجب ملاحظة أن مقارنة القيمة الحرفية (String) تبدأ من أول حرف وحتى نهاية السلسلة (String) . فإذا كانت حروف السلسلة مطابقة لأول حروف في الحقل المفهرسي ، فإن الحقل يعتبر مطابقاً بصرف النظر عن باقي حروفه ، وفي هذه الحالة يقف مؤشر السجلات (Record Pointer) عند أول سجل مطابق . وإذا لم يتطابق الحقل مع القيمة الحرفية المطلوبة تظهر الرسائل "No found" . وفي هذه الحالة ينتقل المؤشر إلى نهاية الملف .

وللبحث عن تاريخ معين يجب أولاً تحويله من حروف (Characters) إلى تاريخ (Date) وذلك باستخدام الدالة (CTOD) كالآتي مثلاً :

SEEK CTOD ('01/01/90')

## ٩١ - الأمر (SELECT)

يستخدم هذا الأمر في التعامل مع مناطق العمل المختلفة (Work Areas) حيث يمكن عن طريق هذا الأمر فتح حتى عشر مناطق عمل ، وهذا يتيح التعامل مع عشرة ملفات قواعد بيانات في نفس الوقت . والصورة العامة لهذا الأمر كالآتي :

SELECT <work area/alias>

ومناطق العمل (Work Areas) تأخذ الأرقام من ( ١ ) إلى ( ١٠ ) أو الحروف من (A) إلى (J) . كما يمكن استخدام المرادفات (Aliases) في تسمية مناطق العمل والملفات الموجودة فيها . ويراعى عند اختيار المرادفات ألا تكون الحروف من (A) إلى (J) . وذلك لأن هذه الحروف تمثل المرادفات المبدئية (Default) لمناطق العمل من ( ١ ) إلى ( ١٠ ) . ولكن يمكن الجمع بين الحروف لتكوين مرادفات مثل (AA) أو (BB) وهكذا .

ويمكن فتح عدة ملفات قواعد بيانات في مناطق عمل مختلفة . ولكن لا يتم التعامل إلا مع آخر منطقة عمل تم فتحها باستخدام الأمر (SELECT) .

ويمكن عرض بيانات من مناطق عمل أخرى غير منطقة العمل الحالية (Current Work Area) ، وذلك باستخدام العلامة (->) . وهذه العلامة تتكون من حرف ناقص (-) يليه حرف أكبر من (>) . فعند كتابة هذه العلامة بعد اسم منطقة العمل أو المرادف (Alias) ، يتم الحصول على البيانات الموجودة في الحقل الخاص بالسجل الذي يقف عنده المؤشر (Pointer) . وذلك كالآتي :

Alias -> fieldname

ويتم ذلك عادة بعد ربط الملفات الموجودة في مناطق عمل مختلفة باستخدام الأمر (SET RELATION) . ويجب ملاحظة أن كل منطقة عمل لها مؤشر سجلات (Record Pointer) خاص بها . كما أن مؤشر السجلات الوحيد الذي يمكن التحكم فيه هو مؤشر منطقة العمل الحالية (Current Work area) أما باقي المؤشرات فتظل ساكنة . وذلك باستثناء استخدام الأمر (SET RELATION) ، فإنه يؤدي إلى ربط جميع مؤشرات مناطق العمل المفتوحة ، بالمؤشر الموجودة في منطقة العمل الحالية .

## أمثلة

عندما يراد ربط ملف الطلبة (Cadets1) بملف الطلبة (Cadets2) يتم فتح

كل ملف في منطقة عمل مختلفة كالآتي :

```
SELECT 1
USE Cadets1
SELECT 2
USE Cadets2 INDEX Name
SELECT 1
SET RELATION TO name INTO cadets2
```

وبلاحظ في هذه الحالة أن منطقة العمل الحالية (Current) هي المنطقة ( ١ ). كما أن الملف الذي يتم ربطه بالملف الموجود في منطقة العمل الحالية يجب أن يكون مفهرسا بناء على حقل مشترك بين الملفين .

## ٩٢ - الأمر (SET)

يستخدم هذا الأمر في عرض قائمة تتضمن بعض الاختيارات التي تساعد المستخدم على التحكم في بيئة الحاسب (Environment) والصورة العامة له كالآتي :

SET

وعند كتابة هذا الأمر أمام مشيرة النقطة يظهر عمود الاختيارات الذي يتضمن الاختيارات الآتية : انظر الشكل ( ٢٧ - ١٩ )

Options	Screen	Keys	Disk	Files	Margin	Decimals
Alternate	OFF					
Bold	ON					
Copy	OFF					
Catalog						
Century	OFF					
ConFirm	OFF					
Deleted	OFF					
Delimiters	OFF					
Device	SCREEN					
Enhistory	OFF					
Escape	ON					
Exact	OFF					
Fielda	OFF					
Fixed	OFF					
Heading	ON					
Help	ON					
History	ON					
Intensity	ON					

شكل ( ٢٧ - ١٩ )



## أ - الاختيار (Options)

ويحتوي هذا الاختيار على اختياريين . الاختيار الأول هو المساعدة (Help) وهو يؤدي إلى عرض شاشات المساعدة التي تظهر مع بعض الأوامر ، وتقوم بإرشاد المستخدم إلى وظائف المفاتيح المختلفة . ويمكن كتابة الأمر الذي يؤدي هذه الوظيفة من مشيرة النقطة (Dot Prompt) ويكون كالآتي : (SET HELP ON) . كما يمكن إيقاف هذه الشاشات عن طريق كتابة الأمر (SET HELP OFF) .

والاختيار الثاني هو (Device) ويتم من طريقه توجيه المخرجات إلى الشاشة (Screen) أو الطابعة (Printer) . ويمكن كتابة الأمر الذي يؤدي هذه الوظيفة من مشيرة النقطة (Dot Prompt) . ويكون كالآتي (SET DEVICE TO SCREEN) عندما يراد توجيه المخرجات إلى الشاشة ، أو كالآتي (SET DEVICE TO PRINT) عندما يراد توجيه المخرجات إلى الطابعة .

## ب - الاختيار (Screen)

ويستخدم هذا الاختيار للتحكم في ألوان الشاشة واختيار الألوان المطلوبة للخلفية (Background) والأعمدة الضوئية (Highlights) وذلك من خلال قائمة الاختيارات الفرعية الخاصة به . ويمكن التحكم في الألوان أيضا عن طريق مشيرة النقطة (Dot Prompt) وذلك بكتابة الأمر (SET COLOR TO) .

## ج - الاختيار (Keys)

ويستخدم هذا الاختيار في تغيير وظائف المفاتيح المساعدة (Function Keys) التي تظهر في قائمة الاختيارات الفرعية والموضح بها وظائف هذه المفاتيح مع ملاحظة أن المفتاح (F1) يكون محجوزا للمساعدة (Help) . وباقي المفاتيح من (F2) إلى (F10) يتم تغيير وظائفها حسب الحاجة . ويمكن استخدام عددا من الحروف حتى (٣٠) حرفا تمثل وظيفة كل مفتاح . ويمكن تنفيذ هذه العملية من مشيرة النقطة (Dot Prompt) عن طريق الأمر (SET FUNCTION) .

## مثال

لتخصيص المفتاح (F8) للأمر (CLEAR) من خلال مشيرة النقطة يتم كتابة السطر التالي :

SET FUNCTION 8 To 'CLEAR'

د - الاختيار (Disk)

ويستخدم هذا الاختيار في معرفة وحدة الأقراص البدئية (Default) وتحديد وحدة الأقراص التي يتعامل البرنامج مع الملفات الموجودة فيها . ويمكن تنفيذ هذه العملية من خلال مشيرة النقطة عن طريق الأمر (SET DEFAULT TO) .

هـ - الاختيار (Files)

ويستخدم هذا الاختيار لفتح ما يسمى بالملف البديل (Alternate File) وهو ملف نصوص (Text File) يستخدم في تخزين الأوامر التي يتم كتابتها من مشيرة النقطة (Dot Prompt) .

ويمكن فتح هذا الملف من مشيرة النقطة باستخدام الأمر (SET ALTERNATE TO) .

و - الاختيار (Margin)

يستخدم هذا الاختيار في ضبط الهامش الأيسر في التقارير وكذلك في حقول الملاحظات (Memo Fields) . ويمكن تنفيذ نفس العملية من خلال مشيرة النقطة عن طريق الأمر (SET MARGIN TO) .

ز - الاختيار (Decimals)

ويستخدم هذا الاختيار في تحديد عدد الكسور العشرية المطلوب ظهورها في الأعداد . ويمكن تنفيذ نفس هذه العملية من خلال مشيرة النقطة عن طريق كتابة الأمر (SET DECIMALS TO) .

٩٢ - الأمر (SET BELL)

يستخدم هذا الأمر في التحكم في الصوت الذي يصدر عند ادخال أي بيانات خطأ، أو عند امتلاء العمود الضوئي المثل للحقل بالحروف عند ادخال البيانات . والصورة العامة له كالآتي :

SET BELL ON/OFF

والوضع المبدئي (Default) هو ON .

## ٩٤ - الأمر (SET CATALOG)

يستخدم هذا الأمر في فتح أو إغلاق ملف الكتالوج . ولف الكتالوج هو ملف يحتوى على جميع ملفات قواعد البيانات والملفات المرتبطة بها ، والتي يتم انشاؤها داخل هذا الكتالوج ، وهو يشبه استخدام الفهارس (Directories) والفهارس الفرعية (Subdirectories) في نظام التشغيل (MS-DOS) . والصورة العامة لهذا الأمر كالآتي :

SET CATALOG ON/OFF

وعند كتابة الأمر (SET CATALOG ON) فان أى ملفات قواعد بيانات يتم انشاؤها تضاف الى الملفات المخزنة في الكتالوج المفتوح . أما عند كتابة الأمر (SET CATALOG OFF) ، فان أى ملفات جديدة يتم انشاؤها لاثضاف الى هذا الكتالوج .

ويجب ملاحظة أن الأمر (SET CATALOG OFF) لايفلق الكتالوج ولكنه يمنع اضافة أى ملفات جديدة اليه ، وذلك عكس الأمر (SET CATALOG TO) دون كتابة أى شيء بعد الأمر ، فان هذا يؤدي الى إغلاق ملف الكتالوج وبالتالي عدم القدرة على التعامل مع الملفات المخزنة فيه .

## ٩٥ - الأمر (SET CATALOG TO)

يستخدم هذا الأمر في فتح كتالوج معين . وهو يختلف عن الأمر (SET CATALOG ON) في أنه يفتح كتالوج محدد باسمه . أما الأمر (SET CATALOG ON) فانه يؤدي الى تجهيز الكتالوج المفتوح لاستقبال أى ملفات أخرى يتم اضافتها . والصورة العامة لهذا الأمر كالآتي :

SET CATALOG TO <filename>

والأمر بهذه الصورة يؤدي الى فتح الكتالوج الذي يحمل الاسم (filename) اذا كان موجودا ، وإذا لم يكن موجودا يتم انشاء كتالوج جديد بهذا الاسم . واسم الملف يتم كتابته بدون الامتداد حيث أن البرنامج يضيف الامتداد (.cat) .

ويستخدم الأمر (SET CATALOG TO) بدون أى شيء بعده فى إغلاق ملف الكتالوج المفتوح ، وفى هذه الحالة لا يمكن التعامل مع الملفات الموجودة فى هذا الكتالوج . وذلك عكس الأمر (SET CATALOG OFF) الذى لا يغلّق ملف الكتالوج ولكنه يمنع إضافة أى ملفات إليه فقط .

ويمكن استخدام علامة الاستفهام مع هذا الأمر كالآتى :

#### SET CATALOG TO?

وفى هذه الحالة تظهر قائمة بأسماء ملفات الكتالوج الموجودة على القرص الحالى (Current Drive) .

وعند استخدام ملف الكتالوج تصبح منطقة العمل العاشرة محجوزة لهذا الكتالوج . وبذلك يصبح متاحا فقط تسع مناطق عمل (Work Areas) للتعامل مع الملفات من خلالها .

#### ٩٦ - الأمر (SET CENTURY)

يستخدم هذا الأمر فى عرض التاريخ أو إدخاله متضمنا الأرقام الدالة على القرن مثل (1955) مثلا بدلا من (55) فقط .

والصورة العامة له كالآتى :

#### SET CENTURY ON/OFF

والوضع المبدئى لهذا الأمر هو (OFF) أى عدم ظهور الأرقام الدالة على القرن .

ويجب ملاحظة أن عرض هذه الأرقام الإضافية لا يؤثر على حجم الحقل الممثل للتاريخ ، حيث أن حجم حقل التاريخ يكون دائما ثمانية حروف (Bytes) .

مثال

- . ? DATE()  
15/02/90
- . SET CENTURY ON
- . ? DATE()  
15/02/1990

## ٩٧ - الأمر (SET COLOR)

يستخدم هذا الأمر في مساعدة مخطط البرامج على التحكم في ألوان الشاشة .  
وهذا الأمر له صورتان يمكن تلخيصهما كالآتي :

### ١- الصورة الأولى

#### SET COLOR ON/OFF

وتستخدم هذه الصورة في التحويل بين الشاشة الملونة (Colored Monitor) والشاشة الأحادية اللون (Monochrom Monitor) . والوضع المبدئي لهذا الأمر هو الوضع الذي يتم تشغيل برنامج (DBase III+) من خلاله . فإذا بدأ التشغيل على شاشة أحادية اللون (Monochrom) ، يصبح الوضع المبدئي للأمر هو (OFF) . وإذا بدأ التشغيل على شاشة ملونة يصبح الوضع المبدئي للأمر هو (ON) .

### ب - الصورة الثانية

SET COLOR TO [<standard>][,<enhanced>][,<border>]  
[,<background>]

والاختيار (standard) يقصد به لـون الشاشة الخارجية (standard display) .

والاختيار (enhanced) المقصود به لون أى أعمدة ضوئية (Highlights) تظهر على الشاشة .

والاختيار (Border) المقصود به الحدود الخارجية للشاشة .

والاختيار (Background) المقصود به لون الخلفية التي تظهر خلف الحروف . والألوان في كل حالة من هذه الحالات يتم تمثيلها بحروف يمكن تلخيصها في الجدول التالي :

اللون	الحرف	اللون	الحرف
أسود	N	أحمر	R
أزرق	B	بنفسجي	RB
أخضر	G	بنّي	GR
سماوي	BG	أبيض	W
فراغ	X		

وتستخدم علامة ( \* ) مع أي لون لجعله يتألق (Blinking) . كما تستخدم علامة الجمع (+) مع أي لون للحصول على اللون الأشد إضاءة (High Intensity). فمثلاً للحصول على اللون الأصفر ، وهو غير موجود في الجدول ، يتم كتابة (GR+) وهو يعني اللون البنّي الشديد الإضاءة أي الأصفر .

## مثال

للحصول على شاشة تحتوي على حروف صفراء على خلفية حمراء على أن يكون لون الحروف داخل الأعمدة الضوئية أبيض على خلفية زرقاء مع حدود خضراء ، يتم كتابة السطر التالي :

SET COLOR TO GR+/R, W/B , G

## ٩٨ - الأمر (SET CONFIRM)

يستخدم هذا الأمر في شاشات الإدخال لمساعد مخطط البرامج على التحكم في انتقال المؤشر (Cursor) من حقل إلى آخر ، حيث أن الوضع المبدئي (Default) أن ينتقل هذا المؤشر إلى الحقل التالي بمجرد امتلاء الحقل بالحروف . فإذا أريد عدم انتقال المؤشر بعد امتلاء الحقل ، يستخدم هذا الأمر . وفي هذه الحالة لا يتم الانتقال إلى الحقل التالي إلا بالضغط على مفتاح الإدخال . والصورة العامة لهذا الأمر كالآتي :

SET CONFIRM ON/OFF

## ٩٩ - الأمر (SET CONSOLE)

يستخدم هذا الأمر في التحكم في الشاشة من خلال البرنامج عن طريق فتحها أو

إغلاقها . والصورة العامة له كالآتي :

SET CONSOLE ON/OFF

وفيجد هذا عندما يراد طباعة التقارير دون ظهورها على الشاشة . ويجب مراعاة إعادة الأمر إلى الوضع المبدئي (Default) وهو (ON) قبل نهاية البرنامج .

### ١٠٠ - الأمر (SET DATE)

ويستخدم هذا الأمر في تغيير صورة التاريخ حسب النظم المختلفة والصورة العامة له كالآتي :

SET DATE AMERICAN/ANSI/BRITISH/ITALIAN/FRENCH/GERMAN

والوضع المبدئي (Default) هو التاريخ الأمريكي (AMERICAN) . والطرق المختلفة لكتابة التاريخ يمكن تلخيصها كالآتي :

AMERICAN	=	MM/DD/YY
ANSI	=	YY.MM.DD
BRITISH	=	DD/MM/YY
ITALIAN	=	DD-MM-YY
FRENCH	=	DD/MM/YY
GERMAN	=	DD-MM-YY

### مثال

لإدخال تاريخ معين في متغير ذاكرة ثم تحويله إلى الشكل الانجليزي (BRITISH) ، يتم كتابة السطور التالية :

```
. mdate = CTOD('01/20/90')
. SET DATE BRITISH
. ? mdate
20/01/90
```

### ١٠١ - الأمر (SET DEBUG ON)

يستخدم هذا الأمر عند اختبار البرنامج لاكتشاف الأخطاء التي قد تكون موجودة به . وهو يرسل خطوات تشغيل البرنامج كما ينفذها الحاسب الى الطابعة . والصورة العامة له كالآتي :

SET DEBUG ON/OFF

وعند كتابة الأمر (SET DEBUG ON) ، يتم توجيه مخرجات الأمر (SET ECHO ON) الى الطابعة بدلا من ظهورها على الشاشة .

### ١٠٢ - الأمر (SET DECIMALS)

يستخدم هذا الأمر لتحديد عدد الأرقام العشرية المطلوب عرضها بالنسبة للمدخلات العددية . والصورة العامة له كالآتي :

SET DECIMALS TO <numeric expression>

والوضع المبدئي (Default) هو ظهور رقمين عشريين في العدد .

### ١٠٣ - الأمر (SET DEFAULT TO)

يستخدم هذا الأمر للانتقال الى وحدة اقراص اخرى لتحميل ملفات أو برامج موجودة عليها . والصورة العامة له كالآتي :

SET DEFAULT TO <drive>

### مثال

عندما يراد تحميل برنامج الطلبة (Cadets) من وحدة الأقراص (B) يتم كتابة السطور التالية :

SET DEFAULT TO B  
USE Cadets



## ١٠٤ - الأمر (SET DELETED)

يستخدم هذا الأمر ليتيح لمخطط البرامج التعامل مع السجلات التي تم وضع علامات عليها لمسحها ، حيث يمكنه عزل هذه السجلات بحيث لا تؤثر فيها الأوامر التي يتم ادخالها . والصورة العامة له كالآتي :

SET DELETED ON/OFF

وعند كتابة الأمر (SET DELETED ON) فان البرنامج يتعامل مع هذه السجلات كأنها غير موجودة .

## ١٠٥ - الأمر (SET DEVICE)

يستخدم هذا الأمر في توجيه السطور التي يتم كتابتها بواسطة الأمر (@...SAY) الى الشاشة أو الطابعة حسب الحاجة . والصورة العامة له كالآتي :

SET DEVICE TO PRINT/SCREEN

والوضع المبدئي (DEFAULT) هو توجيه المخرجات الى الشاشة . وعند كتابة الأمر (SET DEVICE TO PRINT) فان مخرجات الأمر (@...SAY) يتم توجيهها الى الطابعة . أما مخرجات الأمر (@...GET) فانها لا تذهب الى الطابعة .

## ١٠٦ - الأمر (SET DOHISTORY)

يستخدم هذا الأمر للتحكم في تخزين الأوامر في مخزن التاريخ (History) أو عدم تخزينها فيه حسب الحاجة . والصورة العامة له كالآتي :

SET DOHISTORY ON/OFF

ويجب ملاحظة أن هذا الأمر يؤثر فقط في الأوامر التي يتم كتابتها في البرنامج ، ولكنه لا يؤثر في الأوامر المباشرة التي يتم ادخالها عن طريق مشيرة النقطة (Dot Prompt) ، حيث أن هذه الأوامر يتم تخزينها في مخزن التاريخ (History) دون الحاجة الى كتابة هذا الأمر . ويستخدم هذا الأمر عادة عند اختبار البرنامج وتصحيحه .

### ١٠٧ - الأمر (SET ECHO)

يستخدم هذا الأمر في عرض أوامر البرنامج أثناء تنفيذها . والصورة العامة له كالآتي :

SET ECHO ON/OFF

والوضع المبدئي (Default) هو (Off) أى عدم ظهور الأوامر أثناء تنفيذها . وهو يستخدم عادة عند اختبار البرنامج وتصحيحه .

### ١٠٨ - الأمر (SET ESCAPE)

يستخدم هذا الأمر للتحكم فى إيقاف البرنامج أو استمراره نتيجة للضغط على مفتاح الهروب (ESC) . والصورة العامة له كالآتي :

SET ESCAPE ON/OFF

والوضع المبدئي هو (ON) . وهذا يعنى أن الضغط على مفتاح الهروب يؤدي الى إيقاف تنفيذ البرنامج .

### ١٠٩ - الأمر (SET EXACT ON)

يستخدم هذا الأمر فى التحكم فى مقارنة البيانات الحرفية حيث أن المقارنة فى الوضع المبدئي (Default) تتم بين حروف السلسلة الحرفية الأولى والحروف المقابلة لها من السلسلة الحرفية الثانية ، فإذا انتهت السلسلة الأولى أصبحت السلسلتان متطابقتين بالرغم من أنهما فعليا قد يكونان غير متطابقتين تماما . ولذلك يستخدم هذا الأمر فى تحويل المقارنة الى مقارنة تامة (Exact) أى تطابق السلسلتين تماما . والصورة العامة لهذا الأمر كالآتي :

SET EXACT ON/OFF

مثال

عند كتابة السطر التالى :

? 'abc' = 'abcdef'

يظهر على الشاشة النتيجة (.F.) أى غير صحيح (False) .  
وعند كتابة السطر التالى :

? 'abcdef' = 'abc'

يظهر على الشاشة النتيجة (.T.) أى صحيح (True) ، وذلك لأن البرنامج يقارن بين السلسلتين حتى انتهاء السلسلة اليمنى .  
وعند كتابة السطرين التاليين :

SET EXACT ON

? 'abcdef' = 'abc'

يظهر على الشاشة القيمة (.F.) أى غير صحيح (False) . وذلك لأن البرنامج يطابق السلسلتين مطابقة كاملة .

## ١١٠ - الأمر (SET FIELDS)

يستخدم هذا الأمر عندما يراد التحكم فى الحقول التى سبق تحديدها بواسطة الأمر (SET FIELDS TO) ، حيث يتيح لمخطط البرامج استخدام هذه الحقول أو عدم استخدامها حسب الحاجة . والصورة العامة له كالآتى :

SET FIELDS ON/OFF

والوضع البدئى (Default) هو (OFF) ، وهو يعنى أن جميع حقول قاعدة البيانات يمكن التعامل معها ، حتى لو سبق تحديد بعض الحقول باستخدام الأمر (SET FIELDS TO) . أما باستخدام الأمر (SET FIELDS ON) فإن الحقول التى سبق تحديدها فقط هى التى يمكن استخدامها ، وفى هذه الحالة يمكن التعامل مع الحقول الموجودة فى منطقة العمل الحالية (Current work area) أو التعامل مع الحقول الموجودة فى مناطق العمل الأخرى فى نفس الوقت .

## ١١١ - الأمر (SET FIELDS TO)

يستخدم هذا الأمر في تحديد الحقول المطلوب التعامل معها في ملف أو أكثر من ملفات قاعدة البيانات . والصورة العامة له كالآتي :

SET FIELDS TO [<field list>/ALL]

وعندما يراد إضافة حقول من مناطق عمل أخرى غير منطقة العمل الحالية (Current work area) يستخدم الاسم المرادف (Alias) الخاص بهذه المنطقة يليه العلامة (>) التي يتم تكوينها من علامة ناقص (-) وعلامة أكبر من (>) ويلى ذلك اسم الحقل المطلوب ادخاله ضمن قائمة الحقول المطلوبة .

### تحذير

عند استخدام الأمر (SET FIELDS ON) دون تحديد الحقول المطلوبة أولا باستخدام الأمر (SET FIELDS TO) ، لا يمكن التعامل مع الحقول . ويلزم في هذه الحالة استخدام الأمر (SET FIELDS OFF) للعودة الى الوضع المبدئي (Default) .

### ملاحظة

كما سبق الايضاح يمكن استخدام الأمر (SET FIELDS TO) في التعامل مع حقول من ملفات في عدة مناطق عمل في نفس الوقت ولكن هذه الحقول لا تكون مرتبطة ببعضها ، أى لا يمكن عرض بيانات سجل واحد مثلا يتكون من عدة حقول من مناطق عمل مختلفة .

والطريقة الوحيدة لربط هذه الحقول هى استخدام الأمر (SET RELATION TO) في ربط الملفات مع بعضها .

## ١١٢ - الأمر (SET FILTER)

يستخدم هذا الأمر في ترشيح أو تصفية قاعدة البيانات بحيث تبدو كأنها لا تحتوى الا على السجلات التي تحقق الشروط الموجودة في المرشح (Filter) . والصورة العامة لهذا الأمر كالآتي :

**SET FILTER TO [FILE <filename> /?][<condition>]**

ويمكن استخدام الأمر (SET FILTER TO) دون كتابة أى شيء بعده وفى هذه الحالة يتم إغلاق المرشح الذى سبق فتحه لملف قاعدة البيانات المفتوح .

والاختيار (<filename> FILE) يستخدم عندما يراد استخدام ملف البحث (Query File) الذى سبق انشاؤه بواسطة الأمر (CREATE/MODIFY QUERY) . وتستخدم علامة الاستفهام (?) بدلا من اسم الملف لمعرض أسماء جميع ملفات البحث الخاصة بملف قاعدة البيانات المفتوح ، وذلك لاختيار الملف المطلوب منها .

والاختيار (Condition) يستخدم فى ادخال الشرط أو الشروط المطلوب استخدامها فى ترشيح قاعدة البيانات .

## مثال

إذا أريد التعامل مع السجلات الخاصة بالطلبة الذين التحقوا بالمعهد بعد ( ١٩٨٥/١/١ ) مثلا . يتم كتابة السطر التالى :

**SET FILTER TO dat\_ent > CTOD "01/01/85"**

حيث (dat\_ent) هو اسم الحقل الذى يمثل تاريخ التحاق الطالب فى المعهد . وفى هذه الحالة يتعامل البرنامج مع قاعدة البيانات كأنها لا تحتوى إلا على بيانات الطلبة الذين يحققون هذا الشرط .

## ١١٣ - الأمر (SET FIXED)

يستخدم هذا الأمر عندما يراد عرض البيانات العددية محتوية على عدد ثابت من الأرقام العشرية . وهذا العدد يكون فى الوضع المبدئى رقمين ، وذلك فى حالة عدم استخدام الأمر (SET DECIMALS) . والصورة العامة لهذا الأمر هى :

**SET FIXED ON / OFF**

## ١١٤ - الأمر (SET FORMAT)

يستخدم هذا الأمر في فتح ملف التشكيل (Format File) المطلوب استخدامه في ادخال البيانات الى ملف قاعدة البيانات . والصورة العامة لهذا الأمر كالآتي :

SET FORMAT TO [<filename>/?]

ويتم كتابة اسم الملف (filename) بدون الامتداد ، حيث أن البرنامج يفترض أن الامتداد (.fmt) في هذه الحالة . ويمكن استخدام علامة الاستفهام (?) بدلا من اسم الملف لعرض أسماء ملفات التشكيل الخاصة بملف قاعدة البيانات المفتوح واختيار الملف المطلوب منها .

ويمكن اغلاق ملف التشكيل باستخدام الأمر (CLOSE FORMAT) أو الأمر (SET FORMAT TO) دون كتابة أى شيء بعده .

## ١١٥ - الأمر (SET FUNCTION)

يستخدم هذا الأمر في تخصيص وظائف لفاتيح الوظائف (Function Keys) . وفي هذه الحالة يتم تغيير وظيفة المفتاح السابقة بالوظيفة الجديدة التي يتم كتابتها . والصورة العامة لهذا الأمر كالآتي :

SET FUNCTION <exp1> TO <exp2> [;]

حيث (exp1) هو الرقم الممثل للمفتاح المطلوب تغيير وظيفته ، و (exp2) هو اسم الوظيفة المطلوب ادخالها ، وحرف الفاصلة المنقوطة (;) يمثل مفتاح الادخال ، وذلك لكي يتم تنفيذ الوظيفة المطلوبة بمجرد الضغط على المفتاح الخاص بها دون الحاجة الى الضغط على مفتاح الادخال .

ويمكن ادخال اسم الوظيفة المطلوبة (exp2) حتى (٢٠) حرفا . ويجب ملاحظة أنه يمكن تغيير وظائف المفاتيح العشرة ما عدا مفتاح (F1) لأنه يكون محجوزا لتشغيل شاشات المساعدة (Help) .

## مثال

لتخصيص المفتاح (F5) للأمر (DISPLAY) يتم كتابة السطر التالي :

SET FUNCTION 5 TO 'DISPLAY;'

ولتخصيص المفتاح (F8) لتنفيذ عدة وظائف ، يمكن كتابة السطر التالي :

SET FUNCTION 8 TO 'CLEAR;USE CADETS INDEX NAME;'

## ١١٦ - الأمر (SET HEADING)

يستخدم هذا الأمر عندما يراد التحكم في عناوين الحقول التي تظهر مع الأوامر (DISPLAY, LIST, SUM, AVERAGE) ، وذلك بعرضها أو عدم عرضها حسب الحاجة . والصورة العامة لهذا الأمر كالآتي :

SET HEADING ON/OFF

والوضع المبدئي لهذا الأمر هو (ON) ، وفي بعض الأحيان يريد مخطط البرامج ادخال العناوين التي يريد استخدامها باستخدام الأمر (@...SAY) ، في هذه الحالة يقوم باستخدام الأمر (SET HEADING OFF) .

## مثال

عندما يراد عرض سجل مثلا من قاعدة بيانات الطلبة يشمل الحقليين (name) ، (address) يتم كتابة السطور التالية :

USE Cadets  
DISPLAY name, address

وعند الضغط على مفتاح الادخال يظهر الآتي :

Record #	name	address
1	Hatem Mahmoud	20 Tahreer street

وإذا أريد عدم عرض عناوين الحقول تستخدم الأوامر التالية :

USE Cadets  
SET HEADING OFF  
DISPLAY name, address

وعند الضغط على مفتاح الإدخال يظهر الآتي :

1                      Hatem Mahmoud                      20 Tahreer street

### ١١٧ - الأمر (SET HISTORY)

يستخدم هذا الأمر في تشغيل مخزن التاريخ (History) أو عدم تشغيله حسب الحاجة . والصورة العامة له كالآتي :

SET HISTORY ON/OFF

والوضع المبدئي لهذا الأمر هو (ON) . وهذا يسمح باسترجاع الأوامر التي سبق إدخالها من مشيرة النقطة (Dot Prompt) باستخدام مفتاح السهم لأعلى ومفتاح السهم لأسفل . وبالتالي يمكن تنفيذ الأمر المطلوب بالضغط على مفتاح الإدخال فقط دون الحاجة إلى كتابة هذا الأمر مرة أخرى .

### ١١٨ - الأمر (SET HISTORY TO)

يستخدم هذا الأمر في تحديد عدد الأوامر التي يمكن تخزينها في مخزن التاريخ . والصورة العامة له كالآتي :

SET HISTORY TO <exp>

حيث (exp) هو عدد الأوامر المطلوب تحديده . والعدد المبدئي (Default) هو عشرون . والعدد المسموح بإدخاله هو أي عدد من صفر إلى ١٦ ألف أمر حسب الذاكرة المتاحة .

### ١١٩ - الأمر (SET INDEX)

يستخدم هذا الأمر في فتح ملفات الفهرس لاستخدامها فسي ترتيب السجلات . والصورة العامة له كالآتي :



SET INDEX TO [<list of index filenames>/?]

ويتم كتابة اسم الملف بدون الامتداد حيث أن البرنامج يضيف الامتداد (.ndx) ألياً . ويمكن إغلاق ملفات الفهرس المفتوحة عن طريق كتابة الأمر (SET INDEX TO) دون كتابة أى شيء بعده . كما يمكن إغلاقها أيضاً باستخدام الأمر (CLOSE INDEX) . وتستخدم علامة الاستفهام (?) بدلاً من اسم الملف لعرض أسماء جميع ملفات الفهرس الخاصة بملف قاعدة البيانات المفتوح .

ويمكن فتح حتى سبعة ملفات فهرس فى نفس الوقت . ويجب ملاحظة أن هذا الأمر يفتح ملف الفهرس فقط ولكنه لا ينشؤه . ويصبح أول ملف فهرسى يتم فتحه هو الملف الرئيسى (Master) . وعند إجراء أى تعديل فى بيانات ملف قاعدة البيانات ، يتم تحديث ملفات الفهرس المفتوحة تبعاً لهذا التعديل .

## مثال

عندما يراد فتح ملفات الفهرس الخاصة بملف الموظفين (Clients) يتم كتابة السطور التالية :

```
USE Clients
SET INDEX TO Job, Name
```

وهذا يعنى أن ملف الفهرس (Job) هو الملف الرئيسى (Master) .

## ١٢٠ - الأمر (SET MARGIN)

يستخدم هذا الأمر لضبط الهامش الشمال للتقارير للطباعة . والصورة العامة له كالآتى :

SET MARGIN TO <exp>

حيث (exp) هو قيمة عددية تمثل هذا الهامش .

## مثال

SET MARGIN TO 10

### ١٢١ - الأمر (SET MEMOWIDTH)

يستخدم هذا الأمر في تحديد عرض بيانات حقل الملاحظات (memo field) عند عرضها على الشاشة أو طباعتها . والصورة العامة له كالآتي :

SET MEMOWIDTH TO <exp>

والوضع البدئي لهذا العرض هو ( ٥٠ ) حرفا .

### ١٢٢ - الأمر (SET MENU)

يستخدم هذا الأمر للتحكم في ظهور أو عدم ظهور شاشات المساعدة التي تظهر للمستخدم لتعريفه بوظائف مفاتيح التحكم في المؤشر لتصحيح المدخلات قبل ادخالها . والصورة العامة لهذا الأمر كالآتي :

SET MENU ON/OFF

والوضع البدئي هو (ON)

### ١٢٣ - الأمر (SET ORDER)

يستخدم هذا الأمر في التحكم في ترتيب ملفات الفهرس المفتوحة وبالتالي تحديد أيها تصبح ملفا رئيسيا (Master) . والصورة العامة له كالآتي :

SET ORDER TO [<exp>]

حيث : (exp) هو رقم صحيح يأخذ القيمة من صفر الى ( ٧ ) حسب عدد ملفات الفهرس المفتوحة . وعند كتابة القيمة صفر يعود الملف الى وضعه الأول دون فهرسة . وذلك دون إغلاق ملفات الفهرس . وعند كتابة أى رقم آخر ، فإن

هذا الرقم يمثل ترتيب ملف الفهرس الذى يصبح هو الملف الرئيسى (Master) .

## مثال

لفتح ملفات الفهرس الخاصة بقاعدة بيانات الموظفين (Clients) يتم كتابة السطرين التاليين :

```
USE Clients
SET INDEX TO Job, Name
```

وعندما يراد استخدام ملف (Name.ndx) كفهرس رئيسى يتم كتابة السطر التالى :

```
SET ORDER TO 2
```

## ١٢٤ - الأمر (SET PATH)

يستخدم هذا الأمر فى تحديد مسار معين للملفات المطلوب استخدامها والصورة العامة له كالآتى :

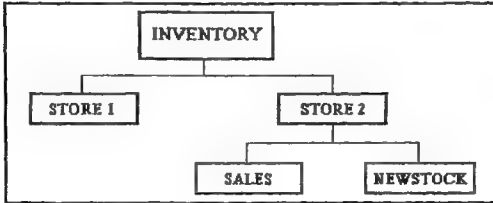
```
SET PATH TO [<path list>]
```

واستخدام الأمر (SET PATH TO) دون كتابة أى شيء بعده يعيد المسار الى الفهرس الفرعى الحالى (Current Directory) . وتحديد المسار يساعد البرنامج على البحث عن الملفات المطلوبة ، وذلك عن طريق تنظيمها على شكل شجرة (Tree) . حيث يبدأ البحث من الجذر (Root) وينتقل الى الفروع والفروع الأصغر وهكذا .

والمسار عبارة عن فهرس فرعية يفصلها الشرطة المائلة (\) ويبدأ بالفهرس الرئيسى (Root Directory) يليه الفهارس الفرعية الأخرى .

وقائمة المسارات (Path List) هى مجموعة من المسارات (Pathes) يفصل بينها فاصلة أو فاصلة منقوطة (;) . وعندما يبحث البرنامج عن أى ملف فإنه يبحث

في المسار الأول . فإذا لم يجده ، ينتقل الى المسار الثاني ثم الثالث ... وهكذا .  
انظر الشكل ( ٢٧ - ٢٠ )



الشكل ( ٢٧ - ٢٠ )

## مثال

إذا كان الملف (Cadets.prg) غير موجود في الفهرس الحالي فان الأوامر التالية تؤدي الى البحث عنه في مسارات أخرى كالآتي :

```
SET PATH TO A:\DBase, B:\Hasan
DO Cadets
```

وفي هذه الحالة يتم البحث في الفهرس الفرعي (\DBase) أولاً ، ثم يتم الانتقال الى الفهرس الفرعي (\Hasan) حتى يتم الوصول الى الملف المطلوب .

## ١٢٥ - الأمر (SET PRINT)

يستخدم هذا الأمر في توجيه المخرجات التي لا يتم كتابتها باستخدام الأمر (SAY...) الى الطابعة مع ظهورها على الشاشة في نفس الوقت ، وهي البيانات الناتجة من استخدام الأوامر (DISPLAY , LIST , ?) . والصورة العامة له كالآتي :

## SET PRINT ON/OFF

والوضع المبني له (OFF) .

## ١٢٦ - الأمر (SET PROCEDURE)

يستخدم هذا الأمر في فتح ملف الخطوات أو الاجراءات (Procedure File) . والصورة العامة له كالآتي :

SET PROCEDURE TO [<filename>]

واسم الملف يجب أن يتضمن رمز وحدة الأقراص إذا كانت غير وحدة الأقراص الحالية (Current Drive) . ولا يشترط اضافة الامتداد حيث أن البرنامج يضيف الامتداد (.prg) آليا .

ويتم اغلاق ملف الخطوات أو الاجراءات عن طريق كتابة الأمر (SET PROCEDURE TO) دون كتابة أى شيء بعده .

وملف الخطوات يمكن أن يحتوى على ( ٢٢ ) برنامج خطوات (Procedure) . ويتم تحديد بداية كل برنامج عن طريق كتابة الأمر (PROCEDURE) في أوله . ويجب ملاحظة أنه لا يمكن فتح أكثر من ملف خطوات في نفس الوقت . وإذا أريد استخدام عدة ملفات ، يتم اغلاق الملف السابق وفتح ملف جديد وهكذا .

## ١٢٧ - الأمر (SET RELATION)

يستخدم هذا الأمر في ربط ملفين من ملفات قواعد البيانات باستخدام حقول مشترك بين الملفين . والصورة العامة له كالآتي :

SET RELATION TO[<keyfield>/<RECNO()> INTO <alias>]

ويقوم هذا الأمر بربط ملف قاعدة البيانات الموجود في منطقة العمل الحالية (Current Work Area) بملف آخر مفتوح في منطقة عمل أخرى . وهذا الملف الآخر يتم تحديده عن طريق الاختيار (alias) أو المرادف الذي يشمل اسم منطقة العمل واسم الملف المفتوح فيها .

ويمكن ربط الملفين عن طريق حقل مشترك فيهما (keyfield) ، وفي هذه الحالة يجب أن يكون الملف مربوط مفهرسا (Indexed) بناء على هذا الحقل المشترك .

كما يمكن ربط الملفين عن طريق رقم السجل (RECNO()) . وفي هذه الحالة يجب أن يكون الملف مربوط غير مفهرس (Unindexed) . ويمكن تخزين هذه العلاقة في ملف منظر (VIEW FILE) ، وذلك عن طريق الأمر (CREATE VIEW FROM ENVIRONMENT) .

## مثال

إذا كان هناك ملفان لبيانات الطلبة (Cadets1) ، (Cadets2) ويراد ربطهما بناء على الحقل المشترك بينهما وهو حقل الاسم (name) يتم كتابة الأوامر التالية :

```
SELECT 1
USE Cadets1
SELECT 2
USE Cadets2 INDEX name
SELECT 1
SET RELATION TO name INTO cadets2
```

يلاحظ أن الملف المطلوب ربطه وهو (Cadets2) تم فهرسته بناء على الحقل المشترك بين الملفين وهو حقل الاسم (name) . كما يلاحظ أن آخر منطقة عمل تم اختيارها بواسطة الأمر (SELECT) هي منطقة العمل التي يتم إنشاء العلاقة داخلها .

## ١٢٨ - الأمر (SET SAFETY)

يستخدم هذا الأمر في تأمين الملفات عن طريق منع النسخ فوقها ، ويحدث هذا عندما يريد المستخدم نسخ ملف باسم ملف موجود . في هذه الحالة تظهر الرسالة التالية :

<filename> already exists, overwrite it ? Y/N

وفي بعض الأحيان يريد مخطط البرامج التحكم في نسخ الملفات دون ظهور هذه

الرسالة التي تسبب توقف البرنامج ، ولذلك يقوم باستخدام هذا الأمر . والصورة العامة له كالآتي :

SET SAFETY ON/OFF

والوضع المبدئي لهذا الأمر هو (ON) .

### ١٢٩ - الأمر (SET SCOREBOARD)

يستخدم هذا الأمر عندما يراد استخدام السطر رقم صفر الذي يكون محجوزا عادة لرسائل برنامج (DBase III+) في حالة استخدام الأمر (SET STATUS OFF) . والصورة العامة له كالآتي :

SET SCOREBOARD ON/OFF

وهو يكون عادة (ON) .

### ١٣٠ - الأمر (SET STATUS OFF)

يستخدم هذا الأمر للتحكم في ظهور عمود الحالة (Status Bar) أو عدم ظهوره . والصورة العامة له كالآتي :

SET STATUS ON/OFF

والوضع المبدئي له (ON) . ويتيح هذا الأمر لمخطط البرامج استخدام السطر رقم ( ٢٢ ) الذي يكون محجوزا لعمود الحالة في عرض بيانات على الشاشة أو تصميم شاشات الإدخال .

### ١٣١ - الأمر (SET STEP)

يستخدم هذا الأمر في تنفيذ خطوات البرنامج خطوة خطوة مع الوقوف بعد كل خطوة ، ويساعد هذا على اكتشاف أخطاء البرنامج ان وجدت . والصورة العامة لهذا الأمر كالآتي :

SET STEP ON/OFF

والوضع الميدني له (OFF) .

## ١٣٣ - الامر (SET TALK)

يستخدم هذا الامر في التحكم في عرض خطوات تنفيذ البرنامج على الشاشة اوعدم عرضها . والصورة العامة له كالآتي :

SET TALK ON/OFF

والوضع الميدني له (ON) . وعادة يحوله مخطط البرامج الى (OFF) عن طريق كتابة الامر (SET TALK OFF) في بداية البرنامج ، حتى يتحكم فيما يعرض على الشاشة أو على الطابعة .

## مثال

يمكن ملاحظة الفرق بين الوضع (ON) والوضع (OFF) مع هذا الامر من خلال السطور التالية :

```
.      STORE 'Mohamed' TO mname1
Mohamed
.      SET TALK OFF
.      STORE 'Hasan' TO mname2
```

يلاحظ بعد السطر الأول ظهور محتويات متغير الذاكرة (mname) نتيجة تنفيذ الامر ، أما بعد السطر الأخير فلا تظهر محتويات متغير الذاكرة نتيجة استخدام الامر (SET TALK OFF) .

## ١٣٣ - الامر (SET TYPEAHEAD)

يستخدم هذا الامر في تحديد عدد الحروف المسموح بالاحتفاظ بها في مخزن الذاكرة (Buffer) قبل انتقالها الى الذاكرة الداخلية . والصورة العامة له كالآتي :

SET TYPEAHEAD TO <exp>



حيث (exp) هو عدد الحروف المطلوب تحديده . والعدد المبدئي يكون عشريين حرفاً . وهذا الأمر يعمل فقط عندما يكون الأمر (SET ESCAPE) على الوضع (ON) . أى أنه لا يعمل عند استخدام الأمر (SET ESCAPE OFF) . وكذلك لا يعمل عند استخدام أى أمر من الأوامر (ON KEY) ، (INKEY) حيث أن هذا يؤدي إلى ادخال أى حرف يكتبه المستخدم إلى الذاكرة الداخلية مباشرة .

### ١٢٤ - الأمر (SET UNIQUE)

يستخدم هذا الأمر للتحكم فى ادخال السجلات التى تشترك فى نفس قيمة الحقل الفهرسى أو عدم ادخالها فى الفهرس . والصورة العامة له كالاتى :

SET UNIQUE ON/OFF

والوضع المبدئي له (OFF)

وعند انشاء ملف فهرس مع استخدام الأمر (SET UNIQUE ON) فان السجلات التى تحتوى على نفس القيمة للحقل الفهرسى لا يتم ادخالها فى الفهرس ، ولكن يتم ادخال أول سجل فقط ، وفى هذه الحالة لا يكون الملف الفهرسى محتويًا على أرقام سجلات مشتركة فى قيمة الحقل الفهرسى . واستخدام هذا الأمر قبل فتح ملف الفهرس (Index File) يؤدي نفس العمل الذى يؤديه الأمر:

INDEX ON <key expression> TO <filename> UNIQUE

وعند اضافة أو تعديل أى سجلات فى الملف فان الملف يستعيد الحالة المنفردة (Unique) ، أى أن اضافة أى سجلات تحتوى على حقل فهرسى يماثل الحقل الفهرسى لسجلات أخرى موجودة فى الملف ، لا يؤدي إلى اضافة أرقام هذه السجلات فى الفهرس باعتبارها سجلات متكررة .

### ١٢٥ - الأمر (SET VIEW TO)

يستخدم هذا الأمر فى فتح ملف المنظر (View File) . والصورة العامة له كالاتى :

SET VIEW TO <filename>/?

ويتم ادخال اسم الملف (filename) بدون الامتداد حيث أن البرنامج يفترض الامتداد (.vue) ، وتستخدم علامة الاستفهام (?) لعرض أسماء جميع ملفات المنظر المخزنة على القرص أو في الكتالوج المفتوح .

## ١٣٦ - الأمر (SKIP)

يستخدم هذا الأمر في نقل مؤشر السجلات (Record Pointer) إلى سجل تال أو سجل سابق خلال ملف قاعدة البيانات المفتوح . والصورة العامة له كالآتي :

SKIP [<exp>]

حيث (exp) هو عدد السجلات المطلوب تحريك المؤشر خلالها . والمؤشر يتحرك إلى السجلات التالية بما يساوي هذا العدد . كما يمكن أن يتحرك إلى السجلات السابقة عند كتابة علامة ناقص (-) قبل العدد (exp) . وإذا لم يتم إضافة أي عدد إلى الأمر ، فإنه يؤدي إلى تحريك المؤشر سجلاً واحداً إلى الأمام .

## أمثلة

لتحريك المؤشر إلى السجل الثاني في ملف قاعدة بيانات الطلبة (Cadets) يتم كتابة الأوامر التالية :

USE Cadets  
SKIP

ولتحريك المؤشر من السجل الثاني إلى السجل رقم ( ١ ) يتم كتابة السطر التالي :

SKIP 7

وللرجوع إلى السجل رقم ( ١ ) يتم كتابة السطر التالي :

SKIP -3

ويمكن تخزين عدد معين في متغير ذاكرة . واستخدام متغير الذاكرة مع الأمر (SKIP) . وذلك كالآتي :

STORE 2 TO mskip  
SKIP mskip

## ١٢٧ - الأمر (SORT)

يستخدم هذا الأمر في إنشاء ملف قاعدة بيانات يحتوى على سجلات الملف الأصلي مرتبة بالترتيب المطلوب تبعا لمحتويات حقل معين أو عدة حقول ، والصورة العامة له كالآتي :

```
SORT [<scope>] TO [<filename>] ON <field1>[/A]/[D]/[C]  
[,<field2>]/[A]/[D]/[C] .... [WHILE <condition>]  
[FOR <condition>]
```

والاختيار (scope) يستخدم لتحديد المدى في الملف المطلوب ترتيبه ، حيث يمكن ترتيب الملف كله (ALL) أو مجموعة من السجلات ابتداء من سجل معين (REST) . والاختيار (filename) هو اسم ملف قاعدة البيانات المرتب الذي سيتم انشاؤه .

والاختيارات (field1) ، (field2) تستخدم لتحديد الحقل أو الحقول التي يتم الترتيب بناء عليها . ويمكن استخدام حتى عشرة حقول في ترتيب الملف ، والاختيار (/A) يستخدم للترتيب تصاعديا (Ascendingly) . ويجب ملاحظة أن الوضع المبني هو الترتيب تصاعديا .

والاختيار (/D) يستخدم للترتيب تنازليا (Descendingly) والاختيار (/C) يستخدم عندما يراد عدم التمييز بين الحروف الكبيرة (Capital) أو الصغيرة (Small) في الترتيب .

كما يستخدم الاختياران (WHILE) ، (FOR) في تحديد شروط البحث التي يتم من خلالها اختيار السجلات المطلوب ترتيبها .

## مثال

يمكن ترتيب ملف الطلبة (Cadets) بناء على حقل المهنة (Job) والاسم (name) كالآتي :

```
SORT ON Job, name TO scadets
```

حيث يصبح الملف (scadets) ملف قاعدة بيانات آخر يحتوى على نفس سجلات الملف الأصلي (Cadets) مرتبة حسب حقل المهنة والاسم على الترتيب .

## ١٢٨ - الأمر (STORE)

يستخدم هذا الأمر فى إنشاء متغير أو عدة متغيرات ذاكرة (memory variables) والصورة العامة له كالآتى :

STORE <exp> TO <memory variable list>

حيث (exp) هي القيمة أو القيم التي يتم تخزينها فى متغيرات الذاكرة . ويمكن تخزين قيمة واحدة فى عدة متغيرات يتم تحديدها من خلال الاختيار (memory variable list) . وهناك طريقة أخرى يمكن عن طريقها إنشاء متغير الذاكرة وذلك كالآتى :

<memory variable> = <expression>

ولكن هذه الطريقة لا تسمح بإنشاء عدة متغيرات فى سطر واحد مثل الأمر (STORE) .

## ملاحظة

الحقل الذى يحمل نفس اسم متغير الذاكرة تكون له الأسبقية عند تنفيذ أى عملية على هذا الحقل . فمثلا إذا كان هناك حقل يحمل الاسم (name) ومتغير ذاكرة يحمل نفس الاسم (name) ، فإن أى عملية يتم إجراؤها على متغير الذاكرة لا تؤثر فيه ولكنها تؤثر فى الحقل (name) فقط .

## مثال

لتخزين القيمة صفر فى عدة متغيرات (a, b, c) يتم كتابة السطر التالى :

STORE 0 TO a, b, c

## ١٣٩ - الأمر (SUM)

يستخدم هذا الأمر في تجميع الحقول العددية في ملف قاعدة البيانات المفتوح والصورة العامة له كالآتي :

```
SUM [<scope>][<fields>][TO <memvar list>]
[WHILE <condition>][FOR <condition>]
```

ويلاحظ أن جميع الاختيارات اختيارية ، أى يمكن كتابة الأمر دون كتابة أى شيء بعده وفى هذه الحالة يتم تجميع جميع الحقول العددية لجميع سجلات قاعدة البيانات .

والاختيار (scope) يتم من خلاله تحديد المدى الذى يتم البحث فيه عن السجلات المطلوب تجميع بيانات الحقول العددية فيها .

والاختيار (fields) يستخدم لتحديد الحقول المطلوب تجميعها .

والاختيار (TO <memvar list>) يستخدم عندما يراد تخزين هذا المجموع فى متغيرات ذاكرة يتوقف عددها على عدد الحقول المطلوب تجميعها .

كما يستخدم الاختياران (WHILE) ، (FOR) للبحث عن السجلات المطلوب تجميع بياناتها العددية .

## مثال

عندما يراد تجميع حقلى الرتب (Salary) ، والساعات (Hours) فى ملف الموظفين (Clients) يتم كتابة السطر التالى :

```
SUM salary , hours TO msalary , mhours
```

## ١٤٠ - الأمر (SUSPEND)

يستخدم هذا الأمر فى إيقاف تنفيذ أوامر البرنامج إيقافاً مؤقتاً . ويمكن كتابة هذا الأمر داخل البرنامج ليساعد مخطط البرامج على إيقاف البرنامج فى المكان الذى يشك فى وجود خطأ فيه ، حيث يستطيع عرض الأوامر السابقة لهذا المكان والمخزنة فى

مخزن التاريخ (History) ، ثم يمكنه العودة مرة ثانية الى تنفيذ أوامر البرنامج عن طريق الأمر (RESUME) .

## ١٤١ - الأمر (TEXT)

يستخدم هذا الأمر في كتابة نصوص كبيرة على الشاشة أو الطابعة . والصورة العامة له كالآتي :

```
TEXT
<text characters>
ENDTEXT
```

ويتم كتابة النص بين (TEXT) ، (ENDTEXT) ويمكن استخدام هذا الأمر في عرض قوائم الاختيارات الكبيرة أو شاشات المساعدة (Help) .

## مثال

يمكن كتابة السطور التالية في البرنامج :

```
TEXT
this is an example of text that is to be
displayed on the screen as it is.
ENDTEXT
```

## ١٤٢ - الأمر (TOTAL)

يستخدم هذا الأمر في تجميع بيانات الحقول العددية لكل السجلات أو لمجموعة من السجلات التي تحقق شروطاً معينة ، ويتم التجميع لكل مجموعة من السجلات تشترك في قيمة الحقل الذي يستخدم كمفتاح (Key Field) لتقسيم هذه المجموعات . والصورة العامة لهذا الأمر كالآتي :

```
TOTAL ON <keyfield> TO <filename>[<scope>]
[FIELDS <fieldlist>][WHILE <condition>]
[FOR <condition>]
```

والاختيار (keyfield) يستخدم في تحديد المجموعات التي يتم تجميع الحقول العددية بها . وهو أحد حقول ملف قاعدة البيانات .

واسم الملف (filename) هو ملف قاعدة بيانات جديد يتم انشاؤه متضمنا مجموع البيانات العددية للمجموعات المختلفة من السجلات .

والاختيار (scope) يستخدم لتحديد مدى السجلات المطلوب البحث خلاله عن السجلات المطلوب ادخالها في الملف الجديد .

والاختيار (FIELDS <field list>) يستخدم لتحديد أسماء الحقول المطلوب ادخالها في الملف الجديد .

ويستخدم الاختياران (FOR) ، (WHILE) في تحديد السجلات المطلوب ادخالها في الملف الجديد حسب شروط البحث التي يتم ادخالها .

ويجب ملاحظة أن ملف قاعدة البيانات المفتوح والذي يتم تجميع بياناته يجب أن يكون مفهرسا (Indexed) أو مفروزا (Sorted) بناء على الحقل المستخدم كمفتاح (Key Field) .

### ملاحظة

تركيب الملف الجديد (Structure) يكون مماثلا تماما لتركيب ملف قاعدة البيانات الأصلي باختلاف واحد وهو عدم نسخ حقول الملاحظات (memo fields) في الملف الجديد .

### مثال

لتجميع بيانات حقول ملف قاعدة بيانات الموظفين (Clients) تبعا لحقل المهنة (Job) يتم كتابة السطور التالية :

```
USE Clients
INDEX ON Job TO Job
TOTAL ON Job TO Job
```

ويلاحظ من السطر الثاني أنه تم استخدام الحقل (Job) كحقل فهرسي لانشاء ملف الفهرس (Job.ndx) .

وفي السطر الثالث تم تجميع بيانات ملف قاعدة البيانات المفتوح وهو الملف (Clients) بناء على حقل المهنة (Job) ، ونتيجة لذلك تم انشاء ملف قاعدة البيانات (Job.dbf) .

وعند عرض بيانات حقل المرتب (Salary) وساعات العمل (Hours) في ملف قاعدة البيانات الجديد ، يتم ذلك كالآتي :

USE Job

LIST salary , hours

يلاحظ في هذه الحالة ظهور الآتي مثلا :

Record	Job	salary	hours
1	Engineer	256570.00	700
1	Technical	189500.00	650

ويلاحظ هنا مجموع مرتبات المهندسين وكذلك الفنيين .

## ١٤٣ - الأمر (TYPE)

يستخدم هذا الأمر في عرض محتويات ملف نصوص (Text file) . والصورة العامة له كالآتي :

TYPE <filename> [TO PRINT]

واسم الملف (Filename) يجب أن يتضمن الامتداد ورمز وحدة الأقراص المخزن عليها إذا كانت غير وحدة الأقراص الحالية (Current drive) .

## ١٤٤ - الأمر (UPDATE)

يستخدم هذا الأمر في تحديث بيانات ملف قاعدة البيانات المفتوح من ملف آخر مفتوح في منطقة عمل أخرى ، ويتم ادخال التعديلات ببطاقة الملفين تبعا لقيمة حقل معين يستخدم كمفتاح للمطابقة . والصورة العامة لهذا الأمر كالآتي :

UPDATE ON <key field> FROM <alias>

REPLACE <field1> WITH <exp1>

[,<field2> WITH <exp2> ...][RANDOM]



حيث (key field) هو الحقل الفهرسي الذي يتم التحديث بناء عليه ويجب أن يكون الملفان مفهرسين (Indexed) ، أو مفروزين (Sorted) بناء على هذا الحقل وذلك لكي يقف المؤشر الخاص بكل ملف على نفس السجل في الملفين مع كل حركة له . و (alias) هو الاسم المرادف الخاص بالملف الآخر المطلوب التحديث منه بالإضافة الى منطقة العمل الخاصة به . ويجب ملاحظة أن الملف المطلوب تعديله يكون مفتوحا وفي منطقة العمل التي تم اختيارها بواسطة الأمر (SELECT) .

والاختيار (REPLACE) يستخدم في استبدال محتويات الحقول التي يراد تعديلها بمحتويات الحقول الموجودة في الملف الآخر المستخدم في التعديل . والاختيار (RANDOM) يستخدم عندما يكون ملف قاعدة البيانات المطلوب تعديله مفهرسا (Indexed) على الحقل الفهرسي (Key Field) وليس مفروزا (Sorted) .

## مثال

عند تحديث بيانات الملف الرئيسي (Master) لقاعدة بيانات المخازن ، وذلك عن طريق ملف المبيعات (Sales) بناء على حقل رقم الجزء (Part\_no) ، يتم كتابة الأوامر التالية :

```
SELECT 2
USE Sales
SELECT 1
USE MASTER INDEX MASTER
UPDATE ON Part_no FROM Sales ;
REPLACE Qty WITH Qty-Sales -> Qty
```

في هذه الحالة يتم استبدال كمية المخزون (Qty) الموجودة في الملف الرئيسي (Master) ، بنفس هذه القيمة (Qty) مطروحا منها قيمة المبيعات الموجودة في حقل الكمية (Qty) الخاص بملف المبيعات (Sales) .

## ١٤٥ - الأمر (USE)

يستخدم هذا الأمر في فتح ملف قاعدة البيانات وملفات الفهرس المرتبطة به ، وإذا كان ملف قاعدة البيانات يحتوي على حقول ملاحظات (memo fields) يتم فتح ملف الملاحظات (.dbt) أليا . والصورة العامة لهذا الأمر كالآتي :

USE [<filename1>/?][INDEX <filename2>]  
[ALIAS <aliasname>]

والاسم (Filename1) هو اسم ملف قاعدة البيانات . ويمكن استخدام علامة الاستفهام (?) فى عرض ملفات قواعد البيانات المخزنة على القرص لاختيار الملف المطلوب منها .

والاسم (filename2) هو ملف الفهرس المرتبط بقاعدة البيانات المفتوحة . ويمكن فتح حتى سبعة ملفات فهرس مع ملف قاعدة بيانات واحد .

ويمكن استخدام الأمر (USE) دون كتابة أى شيء بعده ، وهذا يؤدي الى اغلاق جميع الملفات المفتوحة .

ويستخدم الاسم المرادف (alias name) فى تحديد اسم الملف متضمنا منطقة العمل المفتوحة .

## ١٤٦ - الأمر (WAIT)

يستخدم هذا الأمر فى إيقاف تنفيذ البرنامج والانتظار حتى يضغط المستخدم على أى مفتاح . والصورة العامة له كالآتي :

WAIT [<message>] [TO <memvar>]

حيث (message) هى رسالة يتم عرضها للمستخدم لتنبيهه الى الضغط على أى مفتاح لاستمرار البرنامج .

والاختيار (<memvar> TO) يستخدم عندما يراد تخزين الحرف الذى يضغط عليه المستخدم فى متغير ذاكرة ، ويمكن كتابة الأمر (WAIT) دون كتابة أى شيء بعده ، وفى هذه الحالة يتم عرض الرسالة "Press any key to continue.." وهى الرسالة المبدئية (Default) .

## مثال

لايقاف البرنامج مؤقتا واعطاء المستخدم الاختيار ليستمّر أو ينتهى تشغيل البرنامج ، يتم كتابة السطور التالية :

```
WAIT 'Do you want to continue? (Y/N)' TO mcon
IF UPPER (mcon) # 'Y'
    RETURN
ENDIF
```

## ١٤٧ - الأمر (ZAP)

- يستخدم هذا الأمر في مسح جميع السجلات من ملف قاعدة البيانات المفتوح .  
والصورة العامة له كالآتي :

ZAP

- وهذا الأمر يماثل استخدام الأمر (DELETE ALL) وبعبارة الأمر (PACK) .  
ويتبع ذلك مسح جميع ملفات الفهرس والملفات الأخرى المرتبطة بقاعدة البيانات .



## **الباب الثامن والعشرون**

### **أهم الدوال المستخدمة**



حتى يستطيع مخطط البرامج التحكم في البرنامج وفي قاعدة البيانات ، يحتاج إلى الأسماء بمعظم الدوال (Functions) المستخدمة بواسطة برنامج (DBaseIII+) أو باقي برامج عائلة (DBase) مثل (DBaseIV) ، (FoxBase +) ، (FoxPro) . وفي هذا الباب يتم شرح معظم هذه الدوال مع توضيح وظائفها بالأمثلة كلما أمكن .

كما يستطيع القارئ الرجوع إلى الملحق الموجود في نهاية الكتاب لمعرفة الدوال الإضافية الخاصة ببرامج (DBaseIV) ، (FoxBase +) ، (FoxPro) .

## ملاحظة

القيم الموجودة بين أقواس مربعة ([]) هي قيم اختيارية يستطيع المستخدم إدخالها أو عدم إدخالها حسب الحاجة .

## ١ - الدالة (٤)

تستخدم هذه الدالة في التعويض بالماكرو (Macro Substitution) . ويحدث ذلك عندما يراد التعويض عن قيمة متغير معين في مكان يعامل فيه هذا المتغير كحروف (Characters) . فمثلاً عندما يراد البحث عن المتغير (mname) الذي يحتوي على الاسم (Mohamed) ، فإن كتابة الأمر (FIND mname) لا تؤدي إلى البحث عن الاسم (Mohamed) ولكنها تؤدي إلى البحث عن الحروف (mname) . في حين يمكن كتابة الأمر (FIND &mname) . وفي هذه الحالة يتم البحث عن محتويات المتغير (mname) وهي (Mohamed) . والصورة العامة لهذه الدالة كالآتي :

[.<exp>] &<character variable>

وبلاحظ أن المتغير في هذه الحالة يجب أن يكون متغيراً حرفياً (Character Variable) . ويستخدم الاختيار (<exp>) . عندما يراد إضافة حروف معينة في نهاية المتغير الحرفي . وفي هذه الحالة تستخدم النقطة ( . ) لتحديد نهاية المتغير الحرفي .

فمثلاً لإدخال علامة الضرب (X) داخل سلسلة حرفية مع إضافة أرقام بعد علامة الضرب يمكن كتابة السطور التالية :

```
STORE "X" TO alpha
STORE "15 & alpha.30" TO show
? show
15 X 30
```

يلاحظ في هذه الحالة ظهور العدد (15) ثم علامة الضرب (X) مكان المتغير (alpha) ثم العدد (30) كما كان مكتوبا في السلسلة الحرفية .

وعندما يراد عرض رسالة معينة متضمنة اسم الشخص المطلوب عرض هذه الرسالة عليه ، يتم كتابة السطور التالية :

```
STORE "Mahmoud" TO mname
STORE "Hello & mname" TO greeting
? greeting
```

يلاحظ في هذه الحالة ظهور الرسالة "Hello Mahmoud".

وإذا تغيرت محتويات المتغير (mname) إلى أي اسم آخر ، فإن الرسالة تتضمن الاسم الجديد مثل (Hello Magdy) مثلا .

وإذا أراد مخطط البرامج إعطاء الفرصة للمستخدم لإدخال اسم ملف قاعدة البيانات المطلوب استخدامه، فيمكنه مثلا إنشاء متغير ذاكرة لاسم الملف مثل (dname) ، ثم يقوم المستخدم بإدخال اسم الملف المطلوب في هذا المتغير . وعن طريق دالة الماكرو (٤) يمكن لمخطط البرامج فتح ملف قاعدة البيانات الذي أدخل المستخدم اسمه فيها وذلك كالآتي :

```
USE & dname
```

## ٢ - دالة القيمة المطلقة (ABS)

تستخدم هذه الدالة في الحصول على القيمة المطلقة لقيمة عددية . والصورة العامة لها كالآتي :

```
ABS(<exp>)
```

حيث (exp) هي القيمة العددية المراد إيجاد القيمة المطلقة لها ، ويلاحظ أنها توضع بين قوسين .



وتستخدم هذه الدالة بصفة خاصة عندما يراد إيجاد الفرق العددي بين قيمتين دون الحاجة إلى معرفة أيهما أكبر من الأخرى . كما يلاحظ من السطور التالية :

```
i = 20
j = 80
? ABS(i-j)
```

وفي هذه الحالة يظهر الفرق ( ٦٠ ) موجبا .

### ٣ - الدالة (ASC)

تستخدم هذه الدالة في الحصول على شفرة الآسكي الخاصة بأول حرف من سلسلة حرفية معينة . والصورة العامة لها كالآتي :

```
ASC (<exp>)
```

فمثلا عند كتابة السطر التالي :

```
ASC (Nagy)
```

يظهر العدد (78) الذي يمثل شفرة الآسكي الخاصة بالحرف (N) . وتستخدم هذه الدالة بصفة خاصة في قوائم الاختيارات عندما يراد اختبار الحرف الذي يدخله المستخدم لتنفيذ أحد الاختيارات .

### ٤ - الدالة (AT)

تستخدم هذه الدالة في البحث عن سلسلة حرفية فرعية (Substring) داخل سلسلة حرفية أخرى . وعندما تجدها فانها تعطى عددا يمثل ترتيب بداية هذه السلسلة الفرعية بالنسبة الى بداية السلسلة الحرفية الأخرى . وإذا كانت السلسلة الحرفية التي يتم البحث عنها (Substring) غير موجودة داخل السلسلة الحرفية (String) ، فان هذه الدالة تعطى القيمة صفر (0) . والصورة العامة لهذه الدالة كالآتي :

```
AT(<exp1>, <exp2>)
```

حيث <exp1> هي السلسلة المطلوب البحث عنها داخل السلسلة الأكبر <exp2>

## مثال

يمكن كتابة السطر التالي :

?AT ("is" , "This is a test")

في هذه الحالة يلاحظ ظهور الرقم (3) . وذلك لأن الحروف (is) تبدأ من الحرف الثالث ( بالرغم من تكرارها ابتداء من الحرف السادس ، لأن الدالة (AT) تعطي رقم أول ظهور للسلسلة )

## ٥ - الدالة (BOF)

تستخدم هذه الدالة لاختبار بداية الملف (Beginning of file) . والصورة العامة لها كالآتي :

BOF()

وهي تعطي قيمة منطقية صحيح (True) أو غير صحيح (False) . وتستخدم بصفة خاصة عندما يراد البحث عن سجل معين بفحص الملف عكسياً ، أي من نهاية الملف إلى بدايته . فمثلاً يمكن كتابة السطور التالية :

```
USE Client
GO BOTTOM
DO WHILE .NOT. BOF()
    IF Job = "Teacher"
        ? RECNO ()
    ENDIF
    SKIP -1
ENDDO
```

## ٦ - الدالة (CDOW)

تستخدم هذه الدالة للحصول على اسم اليوم في تاريخ معين . والصورة العامة لها كالآتي :

CDOW(<exp>)

حيث (exp) قد يكون متغير ذاكرة تاريخي أو حقل تاريخي أو تاريخ اليوم الحالي .

فمثلا إذا كان تاريخ اليوم الحالي (DATE()) هو (02/17/90) فلايجاد اسم هذا اليوم يتم كتابة السطر التالي :

? CROW (DATE())

وفي هذه الحالة يظهر الآتي :

Saturday

## ٧ - الدالة (CHR)

وتستخدم هذه الدالة في الحصول على الحروف والأعداد والحروف الخاصة عن طريق معرفة شفرة الآسكي الخاصة بهذه الحروف . والصورة العامة لها كالآتي :

CHR(<exp>)

حيث (exp) هو رقم صحيح من (١) الى (٢٥٥) .

وتستخدم هذه الدالة عندما يراد استخدام بعض الحروف الخاصة التي لا يمكن كتابتها عن طريق لوحة المفاتيح في رسم أشكال معينة على الشاشة ، وكذلك في تشغيل الجرس (Bell) لتحذير المستخدم عند حدوث خطأ معين .

فمثلا لتشغيل الجرس يتم كتابة السطر التالي :

? CHR(7)

ولإضافة رسالة تحذيرية بعد تشغيل الجرس يتم كتابة السطر التالي :

? CHR(7) + "Uncorrect....try again"

## ٨ - الدالة (CMONTH)

تستخدم هذه الدالة في الحصول على اسم الشهر في تاريخ معين . والصورة العامة لها كالآتي :

CMONTH(<exp>)

- حيث (exp) هو متغير ذاكرة أو حقل تاريخي أو تاريخ اليوم الحالي

## مثال

للحصول على اسم الشهر من تاريخ اليوم الحالي (02/17/90) يتم كتابة السطر التالي :

? CMONTH (DATE())

وفي هذه الحالة يظهر اسم الشهر (February) .

ولتحديد اسم الشهر الذي يلي تاريخ اليوم الحالي بستين يوما مثلا ، يتم كتابة السطر التالي :

? CMONTH (DATE() +60)

وفي هذه الحالة يظهر اسم الشهر (April) .

## ٩ - الدالة (COL)

تستخدم هذه الدالة في تحديد رقم العمود (Column) الذي يقف عنده المؤشر (Cursor) . ويحدث ذلك عندما يراد التحكم في مكان المؤشر على الشاشة وعلى الطباعة من خلال البرنامج . والصورة العامة له كالآتي :

COL()

ويساعد ذلك على تحريك المؤشر الى أماكن مختلفة بالنسبة للمكان الحالي للمؤشر فمثلا يمكن كتابة السطر التالي :

@1,COL() + 5 SAY "Enter your name"

في هذه الحالة يتم تحريك المؤشر خمسة أعمدة بعد آخر عمود كان يقف عنده .

## ١٠ - الدالة (CTOD)

تستخدم هذه الدالة في تحويل التاريخ الحرفي من حروف الى تاريخ . والصورة العامة لها كالآتي :

CTOD (<exp>)

وتستخدم عندما يراد مقارنة تاريخ بتاريخ آخر ، أو لتحديد الفترة الزمنية المحصورة بين تاريخ وتاريخ آخر .

### مثال

عندما يراد طباعة تقرير للبيانات الموجودة في ملف قاعدة بيانات الطلبة ، وذلك بالنسبة للطلبة المتحقين بالمعهد ابتداء من تاريخ معين يقوم المستخدم بادخاله وحتى تاريخ معين يقوم بادخاله أيضا ، في هذه الحالة يتم كتابة السطور التالية :

```
STORE SPACE (8) TO start,end
@10,5 SAY " Enter start date " GET start
@12,5 SAY " Enter end date" GET end
READ
strat = CTOD (start)
end = CTOD (end)
CLEAR
REPORT FORM Cadrep ;
FOR date_ent >= start .AND. date_ent <= end
```

ويلاحظ من السطور السابقة أنه تم تحويل التاريخ الذي يدخله المستخدم من حروف الى تاريخ ، وذلك لاستخدامه في تحديد بداية ونهاية البيانات التي تظهر في التقرير .

## ١١ - الدالة (DATE)

تستخدم هذه الدالة في الحصول على تاريخ اليوم الحالي . والصورة العامة لها كالآتي :

DATE()

مع ملاحظة أن التاريخ يظهر على الصورة الأمريكية وهي (mm/dd/yy) إذا لم يتم تغيير صورة التاريخ بواسطة الأمر (SET DATE) أو الأمر (SET CENTURY) كما سبق الإيضاح .

## مثال

للحصول على تاريخ اليوم الحالى يتم كتابة السطر التالى :

? DATE()

وفى هذه الحالة يظهر التاريخ التالى مثلا : (02/20/90) .

ويمكن تخزين تاريخ اليوم الحالى فى متغير ذاكرة لاستخدامه بعد ذلك فى البرنامج وذلك كالتالى :

STORE DATE () TO mdate

## ١٢ - الدالة (DAY)

تستخدم هذه الدالة فى الحصول على العدد الممثل لترتيب اليوم فى الشهر بالنسبة لتاريخ معين . والصورة العامة لها كالتالى :

DAY(<exp>)

حيث (exp) هو متغير ذاكرة تاريخى أو حقل تاريخى أو تاريخ اليوم الحالى مثلا .

## مثال

للحصول على ترتيب اليوم الحالى وهو (02/17/90) بالنسبة للشهر ، يتم كتابة السطر التالى :

? DAY (DATE())

في هذه الحالة يظهر الرقم (17) الذي يمثل ترتيب اليوم في الشهر .

### ١٣ - الدالة (DBF)

تستخدم هذه الدالة في الحصول على اسم ملف قاعدة البيانات المفتوح في منطقة العمل التي تم اختيارها بواسطة الأمر (SELECT) . والصورة العامة لهذه الدالة كالآتي :

DBF()

وفي حالة عدم وجود أي ملف قاعدة بيانات مفتوح ، تعطي هذه الدالة سلسلة حرفية خالية (Null String) .

وتستخدم هذه الدالة عندما يراد معرفة إذا كان هناك ملف قاعدة بيانات (DBF) مفتوح قبل بداية البرنامج أم لا . وفي حالة وجود ملف مفتوح يتم إغلاقه ثم إعادة فتحه في نهاية البرنامج حتى تعود حالة البرنامج إلى وضعها الأصلي ، ولتنفيذ ذلك يتم كتابة السطور التالية :

```
Null = ""
IF NULL < DBF()
  old_file = DBF()
  USE Clients
  -----
  -----
  ----- Commands
  -----
  -----
  USE & old_file
ENDIF
```

وفي هذا البرنامج يتم اختبار الشرط الموجود بعد (IF) للتأكد من وجود ملف قاعدة بيانات مفتوح قبل البرنامج . فإذا كان هناك ملف مفتوح ، يتم تخزين اسمه في متغير الذاكرة (old\_file) ، وذلك حتى يتسنى فتحه بعد ذلك ، ثم يتم إغلاقه عن طريق فتح ملف الموظفين (Clients) وتنفيذ الأوامر المطلوبة على هذا الملف . وقبل نهاية البرنامج يتم فتح الملف الذي كان مفتوحاً مرة ثانية . ويلاحظ هنا استخدام دالة التعميض (&) لأن اسم الملف (DBF) ليس معروفاً ولكنه مخزن في متغير الذاكرة (old\_file) .

## ١٤ - الدالة (DELETED)

تستخدم هذه الدالة في تحديد السجلات التي تم وضع علامات عليها تمهيدا لمسحها . والصورة العامة لها كالآتي :

DELETED()

وهذه الدالة تعطي القيمة صحيح (True) اذا كان السجل الحالي تم وضع علامة عليه لمسح (Marked for deletion)، ويفيد ذلك عندما يريد المستخدم استعراض السجلات التي تم وضع علامات عليها قبل مسحها نهائيا حتى يتأكد أنها السجلات المطلوب مسحها .

فمثلا لعرض بيانات السجلات التي تم تجهيزها للمسح يمكن كتابة السطر التالي :

DISPLAY FOR DELETED

## ١٥ - الدالة (DISKSPACE)

تستخدم هذه الدالة في تحديد حجم الذاكرة المتاح على القرص . والصورة العامة لها كالآتي :

DISKSPACE()

وهي تعطي عددا صحيحا يمثل عدد الحروف (Bytes) المتاحة على القرص . وتفيد عندما يراد عمل نسخ احتياطية من ملف قاعدة البيانات وكذلك عندما يراد عمل فرز للسجلات (Sorting)، حيث أن الفرز يتطلب انشاء ملف قاعدة بيانات جديد بالإضافة الى الملف الأصلي .

فمثلا اذا كان المتغير (mfilesize) يحتوي على عدد الحروف (Bytes) التي يتكون منها ملف قاعدة البيانات ، فيمكن كتابة السطور التالية لاجراء عملية الفرز (Sorting) :

```
IF DISKSPACE() > mfilesize * 2
    SORT ON Last_name TO Newfile
ELSE
    @ 20,15 SAY "There is not enough space"
ENDIF
```



## ١٦ - الدالة (DOW)

- وتستخدم هذه الدالة في الحصول على رقم يمثل ترتيب اليوم في الأسبوع والصورة العامة لها كالآتي :

DOW (<exp>)

حيث (exp) هو متغير ذاكرة تاريخي أو حقل تاريخي أو تاريخ اليوم الحالي ، مع ملاحظة أن الرقم (1) يمثل يوم الأحد (Sunday) . فمثلا للحصول على اليوم الممثل للتاريخ الحالي يتم كتابة السطر التالي :

? DOW (DATE())

- في هذه الحالة يظهر الرقم (7) الممثل ليوم السبت

## ١٧ - الدالة (DTOC)

- تستخدم هذه الدالة في تحويل التاريخ الى حروف . والصورة العامة لها كالآتي :

DTOC (<EXP>)

حيث (exp) هو متغير ذاكرة تاريخي أو حقل تاريخي أو التاريخ الحالي . وتفيد هذه الدالة عندما يراد عرض الحقول التاريخية على الشاشة على صورة التاريخ المعروفة .

## ١٨ - الدالة (EOF)

- تستخدم هذه الدالة في تحديد نهاية ملف قاعدة البيانات (END of file) . والصورة العامة لها كالآتي :

EOF()

وهي تعطي القيمة المنطقية صحيح (True) عندما يصل مؤشر السجلات الى آخر الملف . وهذا لايعني أن المؤشر يكون عند آخر سجل في الملف ولكنه يتخطى هذا السجل ويصل الى علامة نهاية الملف .

وتستخدم هذه الدالة عندما يراد تنفيذ حلقة تكرارية على جميع سجلات قاعدة البيانات .

## مثال

عندما يراد مثلا عرض بيانات الموظفين الذين تزيد أعمارهم عن (٤٠) سنة يتم كتابة الحلقة التكرارية التالية :

```
DO WHILE.NOT.EOF()
  LIST FOR age > 40
ENDDO
```

## ١٩ - الدالة (ERROR)

تستخدم هذه الدالة في الحصول على رقم يحدد الخطأ الذي قد يحدث أثناء تنفيذ البرنامج . والصورة العامة لها كالآتي :

```
ERROR()
```

وتستخدم بصفة خاصة عندما يراد علاج الأخطاء التي قد تحدث أثناء تنفيذ البرنامج ، وذلك عن طريق إغلاق بعض الملفات أو تغيير الأقراص أو مسح بعض الملفات لتوفير مساحة تخزينية . وهكذا .

## مثال

يمكن ادخال السطر التالي في برنامج لاكتشاف أى أخطاء قد تحدث به .

```
ON ERROR DO Err_prg WITH ERROR()
```

فعندما يحدث أى خطأ (ERROR()) يتم تنفيذ البرنامج الفرعى (Err\_prg) مع ادخال الرقم الممثل لهذا الخطأ (ERROR) كعامل (Parameter) للبرنامج الفرعى .

أما البرنامج الفرعى في هذه الحالة فيكون كالآتي مثلا :

```
PARAMETERS Error_no
  IF Error_no =54
  -----
  ----- commands
  -----
  ENDIF
```

حيث الأوامر (Commands) في هذه الحالة تكون رسائل للمستخدم لاتخاذ بعض الإجراءات للتخلص من هذا الخطأ .

## ٢٠ - الدالة (FIELD)

تستخدم هذه الدالة للحصول على اسم أى حقل بمعلومية ترتيب هذا الحقل بين حقول قاعدة البيانات والصورة العامة لها كالآتي :

```
FIELD (<exp>)
```

حيث (exp) هو العدد الذى يمثل ترتيب الحقل ، وهو ينحصر بين ( ١ ) و ( ١٢٨ ) .

## مثال

للحصول على عدد حقول قاعدة البيانات من خلال البرنامج يتم كتابة السطور التالية :

```
USE Clients
num_fields = 0
null= ""
DO WHILE null < FIELD (num_fields + 1)
  num_fields = num_fields + 1
ENDDO
? num_fields
```

في هذا البرنامج يتم تنفيذ الحلقة التكرارية طالما كان اسم الحقل الذى يتم الحصول عليه بواسطة الدالة (FIELD (num\_fields +1)) أكبر من السلسلة

الحرفية الخالية (null string)، وهذا يعني أن الحلقة التكرارية تستمر طالما كان هناك حقول داخل الملف . وفي هذه الحالة يتم زيادة عدد الحقول واحدا . وهكذا يتم تحديد عدد الحقول بعد انتهاء تنفيذ الحلقة التكرارية .

## ٢١ - الدالة (FOUND)

هذه الدالة تعطي القيمة ( صحيح ) أي (True) عندما يصل البرنامج إلى السجل الذي يتم البحث عنه بواسطة الأمر (FIND) أو الأمر (SEEK) أو الأمر (LOCATE) أو الأمر (CONTINUE) . والصورة العامة لها كالآتي :

FOUND()

وتستخدم هذه الدالة في البرنامج عندما يراد تنفيذ بعض الاجراءات في حالة الوصول الى السجل المطلوب أو عدم الوصول اليه .

## مثال

لاستخدام هذه الدالة مع الأمر (LOCATE) يمكن كتابة السطور التالية :

```
LOCATE FOR Job = "Teacher"
DO WHILE FOUND()
    ? name,address
    CONTINUE
ENDDO
```

وفي هذه الحالة يذهب المؤشر الى أول سجل يحقق الشرط ، فاذا وجد أول سجل، يتم تنفيذ الحلقة التكرارية التي يتم عن طريقها عرض بيانات حقول الاسم والعنوان الخاصة بهذا السجل ، ثم يتم البحث من السجل التالي عن طريق الأمر (CONTINUE) . وهكذا يستمر تنفيذ الحلقة التكرارية طالما كان السجل موجودا في كل مرة .

ويمكن استخدام هذه الدالة مع الأمر (SEEK) كالآتي مثلا :

```

SEEK "Teacher"
IF FOUND()
    DO WHILE Job = "Teacher"
        ? name, address
    SKIP
    ENDDO
ENDIF

```

ويراعى فى هذه الحالة أن يكون الملف مفهرسا على حقل الوظيفة (Job) .

## ٢٢ - الدالة (IIF)

تستخدم هذه الدالة لادخال جملة (IF) الشرطية على سطر واحد بدلا من ادخالها على عدة سطور . والصورة العامة لها كالآتى :

IIF(<exp1>,<exp2>,<exp3>)

حيث (exp1) هو الشرط المراد اختباره ، فاذا تحقق فان الدالة تعطي القيمة (exp2) ، واذا لم يتحقق تعطي القيمة (exp3) . وهى تسمى (IF) السريعة ، حيث أنها تؤدي الى سرعة تنفيذ البرنامج وزيادة كفاءته .

## مثال

يمكن ملاحظة الفرق بين استخدام الأمر (IF-ENDIF) واستخدام الدالة (IIF) من خلال هذا المثال ، حيث يتم كتابة السطور التالية التى توضح استخدام الأمر (IF-ENDIF) .

```

IF sex = "F"
    mname = "Ms." + name
ESLE
    mname = "Mr." + name
ENDIF

```

وتنفيذ هذه السطور يؤدي الى تخزين الاسم الموجود فى حقل الاسم (name) فى متغير الذاكرة (mname) ، مسبقا بالحروف (Ms.) اذا كان السجل خاصا بانثى (Female) ، أى أن السجل يحتوى على القيمة (F) فى حقل الجنس (Sex) .

كما يخزنه مسبقاً بالحروف (Mr.) إذا كان السجل خاصاً بذكر في الأحوال الأخرى ، أى السجلات التى تحتوى على أى قيمة أخرى غير (F) . وإذا أريد استخدام الدالة (IIF) لتنفيذ نفس العملية ، يتم كتابة السطر التالى :

```
mname = IIF(sex = "F" , "Ms." , "Mr.") + name
```

وعند تنفيذ هذا السطر ، يتم اختبار الشرط (Sex = "F") فإذا تحقق يتم إضافة الحروف (Ms.) قبل الاسم الموجود فى الحقل (name) وتخزين القيمة الناتجة فى المتغير (mname) . وإذا لم يتحقق ، يتم إضافة الحروف (Mr.) قبل الاسم الموجود فى الحقل name وتخزين القيمة الناتجة فى المتغير (mname) .

## ٢٣ - الدالة (INKEY)

هذه الدالة تعطي القيمة العددية المثلة لآخر حرف تم الضغط عليه بواسطة المستخدم . والصورة العامة لها كالآتى :

```
INKEY()
```

وهي تعطى عدداً صحيحاً بين ( صفر ) و ( ٢٥٥ ) يقابل شفرة الآسكى (ASCII Code) الخاصة بهذا الحرف ، وتستخدم عندما يراد اختبار الحروف التى يضغط عليها المستخدم .

## مثال

السطور التالية توضح استخدام عداد للوقت يحدد الزمن الذى يقضيه المستخدم قبل ادخال الاختيار المطلوب .

```
DO WHILE.T.
-----
-----
----- Menu Options
-----
-----
i = 0
DO WHILE i = 0
    @ 1,72 SAY TIME()
```

```

        i = INKEY()
    ENDDO
    DO CASE
        CASE CHR(i)$ "Aa"
            DO <program1>
        CASE CHR(i)$ "Bb"
            DO <program2>
        CASE CHR(i)$ "Cc"
            DO <program3>
        CASE CHR(i)$ "Qq"
            RETURN
    ENDCASE
ENDDO

```

ويؤدي تنفيذ الحلقة التكرارية الداخلية الى عرض الوقت عن طريق الدالة (TIME()) طالما كانت (i=0). حيث (i) تمثل القيمة العددية للحرف الذي يضبط عليه المستخدم ، لذلك يتم حساب الوقت المستهلك حتى يضبط المستخدم على أى حرف يمثل أحد الاختيارات الموجودة في القائمة .

## ٢٤ - الدالة (INT)

تستخدم هذه الدالة في تحويل القيم العددية الى أعداد صحيحة عن طريق حذف أى كسور عشرية . والصورة العامة لها كالآتي :

INT(<exp>)

## مثال

لتحويل العدد (10.23) الى عدد صحيح يتم كتابة السطر التالي :

? INT(10.23)

ويلاحظ ظهور العدد (10) في هذه الحالة .

## ٢٥ - الدالة (ISALPHA)

تستخدم هذه الدالة في اختبار أول حرف في قيمة معينة فإذا كان حرفا هجائيا فإنها تعطى القيمة صحيح أى (TRUE) ، وإذا كان رقما أو حرفا من الحروف الخاصة، فإنها تعطى القيمة غير صحيح أى (False). والصورة العامة لهذه الدالة كالآتي :

ISALPHA(<exp>)

حيث (exp) هو القيمة التي يتم اختبارها .

### مثال

يمكن كتابة السطر التالي :

? ISALPHA("abc123")

وفي هذه الحالة تظهر القيمة (.T.) أى صحيح . وذلك لأن أول حرف هو الحرف (a) . كما يمكن كتابة السطر التالي :

? ISALPHA("123abc")

وفي هذه الحالة تظهر القيمة (.F.) أى غير صحيح . وذلك لأن أول حرف ليس حرفا هجائيا .

## ٢٦ - الدالة (ISCOLOR)

تعطى هذه الدالة القيمة صحيح أى (True) إذا كان البرنامج يعمل على حالة الألوان (Color Mode) . وتعطى القيمة غير صحيح أى (False) إذا كان يعمل على حالة اللون الأحادي (Monochrome) . والصورة العامة لها كالآتي :

ISCOLOR()

وتنفذ هذه الدالة في إعطاء مخطط البرامج إمكانية التحكم في تصميم البرنامج بجعله يعمل على الشاشة الملونة أو الأحادية اللون حسب بيئة الحاسب المتوفرة .



## مثال

للتحكم في الألوان من خلال البرنامج ، أى استخدام الألوان في حالة الشاشة الملونة (Color Mode) ، واستخدام الأبيض والأسود في حالة الشاشة الأحادية اللون ، يتم كتابة السطور التالية :

```
IF ISCOLOR()
    SET Color TO GR/B,W/R,GR
ELSE
    SET COLOR TO W+
ENDIF
```

## ٢٧ - الدالة (ISLOWER)

تعطى هذه الدالة القيمة صحيح (True) عندما تبدأ القيمة الحرفية التي يتم اختبارها بحرف صغير (Lowercase) كما تعطى القيمة غير صحيح (False) عندما تبدأ بحرف كبير (Uppercase). والصورة العامة لها كالآتي :

```
ISLOWER()
```

## مثال

لاختبار السلسلة الحرفية (abc123) يتم كتابة السطر التالي :

```
? ISLOWER ("abc123")
.T.
```

وبلاحظ هنا ظهور القيمة المنطقية (.T.) أى صحيح .

ولاختبار السلسلة الحرفية (ABC123) يتم كتابة السطر التالي :

```
? ISLOWER ("ABC123")
.F.
```

وبلاحظ هنا ظهور القيمة المنطقية (.F.) أى غير صحيح .

## ٢٨ - الدالة (ISUPPER)

وهي عكس الدالة السابقة . أى أنها تعطي القيمة صحيح أى (True) عندما تبدأ القيمة الحرفية التى يتم اختبارها بحرف كبير (Uppercase)، كما تعطي القيمة غير صحيح أو (False) عندما تبدأ بحرف صغير (Lowercase). والصورة العامة لها كالآتى :

ISUPPER ( )

## ٢٩ - الدالة (LEFT)

هذه الدالة تعطي عددا من حروف السلسلة الحرفية بدءا من اليسار . والصورة العامة لها كالآتى :

LEFT (<exp1>,<exp2>)

حيث (exp1) هو السلسلة الحرفية المراد سحب جزء منها .

و (exp2) هو عدد يمثل عدد الحروف المطلوب استخراجها من اليسار .

وهذه الدالة تشبه الدالة (SUBSTR) مع اختلاف واحد وهو أنها لا تحتاج الى تحديد بداية السلسلة الحرفية المستخرجة ، حيث أنها تبدأ دائما من أول حرف من اليسار .

## مثال

للحصول على الثلاثة حروف الأولى من الاسم (Mohamed) يتم كتابة السطر التالى .

? LEFT ("Mohamed",3)  
Moh

يلاحظ ظهور الحروف (Moh) .

## ٣٠ - الدالة (LEN)

هذه الدالة تعطي عدداً يمثل عدد الحروف الموجودة في سلسلة حرفية (String). والصورة العامة لها كالآتي :

LEN(<exp>)

حيث (<exp>) هو السلسلة الحرفية المراد حساب طولها .

## مثال

لايجاد طول الحقل (name) الذى يحتوى على الاسم (Hatem Zaky) يتم كتابة السطر التالى :

? LEN (name)

25

يلاحظ في هذه الحالة ظهور الرقم (25) مع أن الاسم المذكور يحتوى على عشرة حروف فقط . وذلك لأن الدالة (LEN) تحسب طول الحقل بالكامل متضمناً الفراغات (Spaces) .

ولحساب الطول الفعلى للاسم يتم التخلص من الفراغات (Spaces) الموجودة بعد الاسم باستخدام الدالة (TRIM) ، ثم حساب طول السلسلة الحرفية بعد ذلك . وذلك كالآتي :

? LEN (TRIM(name))

ويلاحظ في هذه الحالة ظهور العدد (10) الممثل للعدد الفعلى للحروف متضمناً المسافة (space) الموجودة بين الاسمين فقط ، حيث أن الدالة (TRIM) قد ازالته المسافات الموجودة آخر الاسم .

## ٣١ - الدالة (LOG)

هذه الدالة تعطي قيمة اللوغاريتم الطبيعى لأى عدد . والصورة العامة لها كالآتي :

LOG (<exp>)

حيث (<exp>) هو العدد المطلوب إيجاد اللوغاريتم الطبيعي له . واللوغاريتم الطبيعي هو الذي يكون أساسه النسبة التقريبية (e) . فمثلا لايجاد اللوغاريتم للعدد (2.718) الذي يمثل النسبة التقريبية (e) يتم كتابة السطر التالي :

```
? LOG (2.718)
1.00
```

## ٣٢ - الدالة (LOWER)

تستخدم هذه الدالة في تحويل الحروف الكبيرة الى حروف صغيرة . والصورة العامة لها كالآتي :

```
LOWER(<exp>)
```

حيث (<exp>) هو السلسلة الحرفية المطلوب تحويلها الى حروف صغيرة ، وتستخدم هذه الدالة بصفة خاصة في الحالات التي يراد فيها التحكم في البيانات التي يدخلها المستخدم ، فمثلا إذا كان المطلوب من المستخدم ادخال بيانات الاسم (name) بحيث تكون بحروف صغيرة حتى تكون كل السجلات متماثلة ، فيتم استخدام هذه الدالة في تحويل البيانات التي يدخلها المستخدم الى حروف صغيرة (Lowercase) . فاذا أدخل المستخدم حروفا صغيرة أو كبيرة يتم تحويلها الى صغيرة .

## مثال

عندما يراد البحث عن اسم معين في قاعدة بيانات الطلبة مثلا يتم كتابة سطور التالية :

```
@ 15,15 SAY "Enter name to look for" ;
GET Lookup
READ
Lookup = LOWER (Lookup)
SEEK Lookup
```

يلاحظ في هذه الحالة تحويل الاسم الذي يدخله المستخدم الى حروف صغيرة قبل

البحث عنه باستخدام الأمر (SEEK). وذلك بفرض أن الأسماء قد سبق تخزينها في قاعدة البيانات بحروف صغيرة .

## ٢٣ - الدالة (LTRIM)

تستخدم هذه الدالة في مسح المسافات الخالية (Spaces) من أول السلسلة الحرفية من اليسار . والصورة العامة لها كالآتي :

`LTRIM(<exp>)`

وتفيد هذه الدالة عندما يراد علاج الأخطاء التي قد تنتج عن ادخال المستخدم لمسافات خالية قبل البيانات التي يقوم بادخالها ، حيث يتم مسح هذه المسافات قبل ادخالها الى الحقول .

فمثلا لكي يدخل المستخدم اسما معيناً في حقل الاسم (name) يتم أولاً انشاء متغير ذاكرة لهذا الحقل مثل (mname). ثم يتم استخدام الدالة السابقة في التخلص من أي مسافات سواء في أول الاسم أو في آخره وذلك كالآتي مثلا :

```
mname = space(30)
@ 10,10 SAY "Enter a name" GET mname
READ
mname = LOWER (LTRIM(TRIM(mname)))
REPLACE name WITH mname
```

في السطر الأول يتم انشاء متغير الذاكرة (mname) .

وفي السطر الثاني يتم عرض رسالة للمستخدم لادخال الاسم.

وفي السطر الثالث يتم تخزين الاسم في المتغير (mname) .

وفي السطر الرابع يتم تحويل الاسم الى حروف صغيرة مع التخلص من المسافات في أول الاسم وآخره باستخدام الدالتين (LTRIM) ، (TRIM) .

وفي السطر الخامس يتم استبدال محتويات حقل الاسم للسجل الحالي بالاسم الموجود في متغير الذاكرة (mname) .

## ٣٤ - الدالة (LUPDATE)

هذه الدالة تعطى تاريخ آخر تحديث ثم اجراؤه للملف . والصورة العامة لها كالآتي :

LUPDATE()

وتتيح هذه الدالة لمخطط البرامج التحكم في تحديث المستخدم للبيانات حتي لا يتم تحديثها عدة مرات ، حيث أن تحديثها عدة مرات قد يؤدي الى ادخال سجلات مكررة أو تجميع بيانات عددية أكثر من مرة مما يؤدي في النهاية الى عدم دقة البيانات .

### مثال

يمكن عن طريق السطور التالية عرض رسالة للمستخدم توضح له آخر تاريخ تم فيه تحديث الملف . ثم تترك له حرية الاختيار بين تحديث البيانات اذا كانت هناك بيانات جديدة مطلوب تحديثها بعد هذا التاريخ ، أو الاكتفاء بالتحديث الذي سبق اجراؤه .

```
IF LUPDATE() < DATE()
  entry = "?"
  @ 5,5 SAY "last entry was on"+DTC(LUPDATE());+
    "Enter now ?(Y/N)" GET entry PICTURE "Y"
  READ
  IF entry = "Y"
    DO <entry program>
  ENDIF
ENDIF
```

ويلاحظ في السطر الثالث استخدام علامة (+) ثم الفاصلة المنقوطة (;) وذلك لربط السلسلة الحرفية في هذا السطر بالسلسلة الحرفية المكتملة لها في السطر التالي . وعند الرغبة في تحديث بيانات الملف يتم ادخال (Y) فيتم تنفيذ البرنامج (entry program) الذي يؤدي الى ادخال البيانات المطلوب تحديثها .

## ٣٥ - الدالة (MAX)

هذه الدالة تعطي أكبر قيمة من قيمتين عدديتين . والصورة العامة لها كالآتي :

`MAX(<exp1>, <exp2>)`

## ٣٦ - الدالة (MIN)

هذه الدالة عكس الدالة السابقة ، حيث تعطي أقل قيمة من قيمتين عدديتين . والصورة العامة لها كالآتي :

`MIN(<exp1>, <exp2>)`

وتستخدم هذه الدالة عندما يراد مثلا الحصول على أقل قيمة عددية لحقل معين .

فمثلا للحصول على أقل مرتب لموظف في قاعدة بيانات الموظفين (Clients) يتم كتابة السطور التالية :

```
USE Clients
min_sal = salary
DO WHILE .NOT. EOF()
    min_sal = MIN(min_sal, salary)
SKIP
ENDDO
USE
RETURN
```

في هذا البرنامج يتم الانتقال الى السجل التالي دائما بواسطة الأمر (SKIP) ، ثم يتم الحصول على أقل قيمة من المرتب (Salary) والمرتب السابق ثم يتم تخزين هذه القيمة في المتغير (min\_sal) ، وهكذا تتكرر هذه العملية حتى يتم الحصول على أقل قيمة للمرتب .

## ملاحظة

يمكن استخدام نفس الطريقة في الحصول على أكبر قيمة بواسطة الدالة (MAX) .

## ٢٧ - الدالة (MOD)

هذه الدالة تعطي باقى القسمة الصحيحة لعدد على عدد آخر . والصورة العامة لها كالآتي :

$$\text{MOD}(\langle \text{exp1} \rangle, \langle \text{exp2} \rangle)$$

وتستخدم هذه الدالة بصفة خاصة فى التحويل من وحدات الى أخرى ، مثل تحويل الياردة الى بوصة والمتر الى سنتيمتر والساعة الى دقائق وثوان . وهكذا .

فمثلا اذا كان هناك عدد من الدقائق يراد تحويله الى عدد من الأيام وعدد من الساعات وعدد من الدقائق ، يمكن كتابة السطور التالية :

```
t = 36500
minutes = MOD(t,60)
h = INT(t/60)
hours = MOD(h,24)
days = INT(h/24)
? t,"minutes are:",days,"days",hours,"hours";
minutes, "minutes"
```

وعند الضغط على مفتاح الادخال بعد السطر الأخير يلاحظ ظهور الآتى :

```
36500 minutes are:25 days 8 hours 20 minutes
```

والسطر الأول من البرنامج يتم عن طريقه انشاء المتغير (t) الذى يحتوى على العدد (36500) دقيقة .

والسطر الثانى يقوم بتحديد عدد الدقائق التى تبقى بعد القسمة على ( ٦٠ ) . وهى الدقائق التى تبقى بعد تحديد الأيام والساعات .

والسطر الثالث يقوم بتحديد عدد الساعات الصحيحة الموجودة الناتجة عن قسمة عدد الدقائق على ( ٦٠ ) .

والسطر الرابع يقوم بتحديد الساعات الباقية بعد القسمة على ( ٢٤ ) .



والسبتر الخامس يحدد عدد الأيام الصحيحة الناتجة عن قسمة الساعات على ( ٢٤ ) .

والسبتر السادس يقوم بعرض البيانات التي تم حسابها على الشاشة .

## ٢٨ - الدالة (MONTH)

- هذه الدالة تعطي عددا يمثل ترتيب الشهر في السنة بالنسبة لتاريخ معين والصورة العامة لها كالآتي :

MONTH(<exp>)

- حيث (exp) هو متغير ذاكرة حرفي أو حقلا حرفيا أو تاريخ اليوم الحالي .

فمثلا إذا كان تاريخ اليوم الحالي (2/18/1990) ، يمكن الحصول على رقم الشهر كالآتي :

? MONTH(DATE())

- في هذه الحالة يظهر الرقم (2) الممثل لترتيب الشهر .

## ٢٩ - الدالة (NDX)

هذه الدالة تعطي اسم ملف الفهرس المفتوح في منطقة العمل (Work Area) التي سبق اختيارها بواسطة الأمر (SELECT) . والصورة العامة لهذه الدالة كالآتي :

NDX(<exp>)

- حيث (exp) هو رقم يمثل ترتيب ملف الفهرس بين الملفات المفتوحة ، وهو يأخذ أي رقم من (1) إلى (7) .

## مثال

لكي يعرف البرنامج أسماء ملفات الفهرس المفتوحة ويتعامل معها يمكن كتابة السطور التالية :

```
i =1
Null= ""
DO WHILE Null < NDX(i).AND.i <= 7
    ? NDX(i)
    i =i +1
ENDDO
```

## ٤٠ - الدالة (OS)

هذه الدالة تعطى اسم نظام التشغيل الذى يعمل عليه البرنامج . والصورة العامة لها كالآتى :

```
OS()
```

وتستخدم هذه الدالة عندما يراد تصميم برنامج نقال (Portable) ، أى يمكنه العمل على نظم تشغيل مختلفة مثل (MS-DOS) ، (UNIX) .

## مثال

لتحويل البرنامج الذى يعمل على نظام التشغيل (MS-DOS) ليعمل على نظام التشغيل (UNIX) ، يتم كتابة السطور التالية :

```
STORE OS() TO opsys
IF SUBSTR(opsys,1,4) = "UNIX"
    DO setunix
ENDIF
```

وتؤدى هذه السطور الى اختبار نظام التشغيل المستخدم فاذا كان النظام (UNIX) يتم تنفيذ البرنامج (setunix) الذى يؤدى الى تجهيز البرنامج للتعامل مع نظام التشغيل (UNIX) .

## ٤١ - الدالة (PCOL)

تستخدم هذه الدالة في تحديد العمود (Column) الذي يقف عنده رأس الطباعة بالنسبة للورقة الموجودة على الطابعة . والصورة العامة لها كالآتي :

PCOL()

ويمكن عن طريق هذه الدالة تحريك رأس الطباعة على الورقة في أعمدة مختلفة بالنسبة للعمود الذي يقف عنده .

فمثلا عندما يراد الطباعة بعد آخر طباعة سبق تنفيذها بخمسة أعمدة يتم كتابة السطور التالية :

```
SET DEVICE TO PRINT
@ 1,PCOL()+ 5 SAY "This is a test"
SET DEVICE TO SCREEN
```

كما يمكن عن طريق هذه الدالة معرفة رقم العمود الذي يقف عنده رأس الطباعة بالنسبة للورقة . كالآتي مثلا :

? PCOL()

في هذه الحالة يظهر العدد (5) مثلا . أى أن رأس الطباعة يقف عند العمود (5) من الورقة ، وهكذا .

## ٤٢ - الدالة (PROW)

تستخدم هذه الدالة في تحديد السطر (ROW) الذي يقف عنده رأس الطباعة بالنسبة للورقة . والصورة العامة لها كالآتي :

PROW()

كما تستخدم أيضا في تحريك رأس الطباعة عددا من السطور بالنسبة لآخر سطر كان يقف عنده .

فمثلا عندما يراد الطباعة بعد آخر طباعة سبق تنفيذها بخمسة سطور  
(PROW()+5) ، يتم كتابة السطور التالية :

```
SET DEVICE TO PRINT
@ PROW()+ 5,1 SAY "This is a test"
SET DEVICE TO SCREEN
```

## ٤٣ - الدالة (RECCOUNT)

هذه الدالة تعطي عدد السجلات (Records) في ملف قاعدة البيانات المفتوح .  
والصورة العامة لها كالآتي :

RECCOUNT()

وتستخدم هذه الدالة بصفة خاصة في البرامج التي يتم من خلالها عمل نسخ احتياطية (Backups) للملفات قواعد البيانات أليا . حيث يتم استخدامها مع الدالة (RECSIZE()) والدالة (DISKSPACE()) في تحديد الحجم المتاح على القرص . وإذا كان يكفي لعمل نسخة احتياطية للملف أم لا .

## مثال

لايجاد عدد السجلات في ملف بيانات الطلبة يتم كتابة السطور التالية :

```
USE cadets
? RECCOUNT
```

في هذه الحالة يظهر العدد الذي يمثل عدد السجلات في الملف .

## ٤٤ - الدالة (RECNO)

تستخدم هذه الدالة في تحديد رقم السجل (Record number) الذي يقف عنده المؤشر . والصورة العامة لها كالآتي :

RECNO()

وتنفذ هذه الدالة بصفة خاصة بعد أوامر البحث مثل الأمر (SEEK) لمعرفة إذا كان هناك سجل يحقق الشرط أم لا .

فمثلا عندما يراد البحث عن اسم معين موجود في المتغير (Lookup) يتم كتابة السطور التالية :

```
SEEK Lookup
Recno = RECNO()
IF Recno>0
    SET FORMAT TO Cadets
    READ
    CLOSE FORMAT
ELSE
    @ 10,10 SAY "There is no & lookup"
    ? CHR(7)
ENDIF
```

وفي هذا البرنامج يتم البحث عن الاسم الموجود في متغير الذاكرة (Lookup) فإذا كان موجودا يتم تخزين رقم هذا السجل في متغير الذاكرة (Recno) . وإذا لم يكن موجودا يتم تخزين القيمة صفر في هذا المتغير . وفي الحالة الأولى يتم فتح شاشة الإدخال عن طريق الأمر (SET FORMAT) ليقيم المستخدم بإدخال البيانات المطلوبة . وفي الحالة الثانية يتم عرض الرسالة المبينة . ويلاحظ أهمية استخدام الماكرو في هذه الرسالة لإظهار محتويات المتغير (Lookup) للمستخدم ليعرف أن هذا الاسم غير موجود .

## ٤٥ - الدالة (RECSIZE)

هذه الدالة تعطي حجم السجل (Record Size) في ملف قاعدة البيانات المفتوح . والصورة العامة لها كالآتي :

RECSIZE()

وتستخدم هذه الدالة بصفة خاصة في البرامج التطبيقية التي يتم من خلالها عمل نسخ احتياطية (Backups) للملفات قواعد البيانات المستخدمة ، وذلك بالاشتراك مع الدالة (RECCOUNT()) والدالة (DISKSPACE()) . حيث تساعد على التأكد من

وجود مساحة متاحة في القرص لتخزين النسخة الاحتياطية .

فمثلا لعمل نسخة من ملف قاعدة بيانات كبير يحتاج الى عدة أقراص لتخزينه، يتم كتابة السطور التالية :

```
USE File1
SET DEFAULT TO B
DO WHILE .NOT. EOF()
    WAIT "Insert new disk in drive B, and press a key."
    COPY NEXT(DISKSPACE() - <header size>)/RECSIZE() ;
    TO Backup
SKIP
ENDDO
USE
```

حيث (header Size) هو حجم العنوان الذي يكون موجودا قبل كل سجل ويتم حسابه من العلاقة الآتية :

$$\text{header size} = 32 * \langle \text{number of fields} \rangle + 35$$

والبرنامج السابق يؤدي الى استمرار نسخ السجلات سجلا سجلا طالما كانت القيمة الناتجة بعد الأمر (COPY NEXT) أكبر من واحد . وعندما تقل هذه القيمة عن واحد ، فان هذا يعني أن المساحة الخالية (Disk Space) المتاحة على القرص أقل من حجم السجل التالي ، وبالتالي يتوقف النسخ ويطلب البرنامج من المستخدم وضع قرص جديد . ثم تتكرر هذه العملية حتى يتم نسخ ملف قاعدة البيانات .

## ٤٦ - الدالة (REPLICATE)

تستخدم هذه الدالة في تكرار حرف معين أو قيمة حرفية معينة عددا من المرات يتم تحديده . والصورة العامة لهذه الدالة كالآتي :

REPLICATE(<exp1>,<exp2>)

حيث (exp1) هو القيمة الحرفية المراد تكرارها .

و (exp2) هو العدد الذي يمثل عدد مرات تكرار هذه القيمة .

ويجب ملاحظة أن عدد الحروف الذي يتكون من عملية التكرار يجب ألا يزيد عن (٢٥٤) حرفاً .

وتستخدم هذه الدالة بصفة خاصة في تكوين أشكال على الشاشة مثل المستطيلات التي يتم تكوين أضلاعها من حروف معينة مثل الحرف (\*) أو عن طريق استخدام أى حروف أخرى يتم تكوينها باستخدام الدالة (CHR()).

فمثلاً للحصول على خط أفقي مكون من تكرار الحرف (\*) عدداً محدداً من المرات يتم كتابة السطر التالي :

```
5,5 SAY REPLICATE("*,20)
```

وفي هذه الحالة يظهر الآتي على الشاشة :

```
.....
```

#### ٤٧ - الدالة (RIGHT)

هذه الدالة تعطي عدداً من الحروف الموجودة في سلسلة حرفية بدءاً من اليمين والصورة العامة لها كالآتي :

```
RIGHT(<exp1> , <exp2>)
```

حيث (exp1) هو القيمة الحرفية المراد استخراج عدد من حروفها .  
و (exp2) هو العدد الذي يمثل عدد الحروف المراد استخراجه . وتستخدم هذه الدالة في حالات كثيرة يراد فيها استخراج جزء من سلسلة حرفية معينة

#### مثال

لاستخراج الحروف الثلاثة الأخيرة من الاسم (Mahmoud) يتم كتابة السطر التالي :

```
? RIGHT("Mahmoud",3)
```

في هذه الحالة يلاحظ ظهور الحروف الثلاثة (oud) .

## ٤٨ - الدالة (ROUND)

هذه الدالة تقوم بتقريب العدد لعدد محدد من الكسور العشرية . والصورة العامة لها كالآتي :

$ROUND(<exp1> , <exp2>)$

حيث (exp1) هو العدد المطلوب تقريبه .  
و (exp2) هو عدد الكسور العشرية المطلوب التقريب إليها .

### مثال

لتقريب العدد (10.765788) لأقرب رقمين عشريين يتم كتابة السطر الآتي :

? ROUND(10.765788 ,2)

وعند الضغط على مفتاح الإدخال يظهر العدد (10.77) .

## ٤٩ - الدالة (ROW)

هذه الدالة تعطي السطر الحالي الذي يقف عنده المؤشر على الشاشة . والصورة العامة لها كالآتي :

ROW()

وتستخدم عندما يراد التحكم في مكان المؤشر وتحريكه عددا من السطور بالنسبة للمكان الذي يقف عنده . فمثلا يمكن كتابة السطر التالي :

@ ROW()+5,3 SAY "Enter your name"

في هذه الحالة تظهر الرسالة المبينة بعد خمسة سطور من آخر سطر وصل إليه المؤشر .



## ٥٠ - الدالة (RTRIM)

تستخدم هذه الدالة في مسح المسافات من نهاية قيمة حرفية معينة . والصورة العامة لها كالآتي :

RTRIM(<exp>)

وهي تماثل الدالة (TRIM()) تماما .

## ٥١ - الدالة (SPACE)

تستخدم هذه الدالة في انشاء متغير ذاكرة يحتوى على عدد معين من الحروف الخالية (blanks) . والصورة العامة لها كالآتي :

SPACE(<exp>)

ويمكن تكوين متغير ذاكرة يحتوى على عدد من الحروف يصل الى (٢٥٤) حرفا .

## مثال

عندما يراد مثلا انشاء متغير ذاكرة (mname) حتى يستطيع المستخدم ادخال الاسم فيه يتم كتابة السطور التالية :

```
mname = SPACE(30)
@5,5 SAY "Enter new name" GET mname
READ
```

## ٥٢ - الدالة (SQRT)

هذه الدالة تعطى الجذر التربيعي للقيمة العددية الموجبة . والصورة العامة لها كالآتي :

SQRT(<exp>)

## مثال

لايجاد الجذر التربيعي للعدد (4) يتم كتابة السطر التالي :

? SQRT(4)

في هذه الحالة يظهر العدد (2.00) .

## ٥٣ - الدالة (STR)

تستخدم هذه الدالة في تحويل القيمة العددية الى سلسلة حرفية (String).  
والصورة العامة لها كالآتي :

STR(<exp>, <length> , <decimal>)

حيث (exp) هو القيمة العددية المراد تحويلها .

و (length) هو عدد الأرقام المراد ظهورها وهو اختياري ، وفي حالة عدم ادخاله يظهر حتى عشرة أرقام .

و (decimal) هو عدد الأرقام العشرية وهو اختياري أيضا ، وفي حالة عدم ادخاله يتم التقريب لأقرب عدد صحيح .

وعند ادخال طول (length) أصغر من عدد الأرقام الصحيحة الموجودة في العدد، فإن البرنامج يعرض مجموعة من حروف النجمة (\*) مكان العدد . وعند ادخال عدد أرقام عشرية (Decimal) أقل من عدد الأرقام العشرية الموجود في العدد ، يتم تقريب الأرقام العشرية الزائدة .

## مثال

لعرض العدد (33.56) كحروف (String) ، يتم كتابة السطر التالي :

? STR(33.56 , 4, 1)

في هذه الحالة يظهر العدد (33.6) .

وبلاحظ هنا عند تحديد الطول (Length) تم حساب نقطة الكسر العشري (Decimal Point) ضمن عدد الأرقام ، فأصبح العدد (4) بدلا من (3) .

## ٥٤ - الدالة (STUFF)

تستخدم هذه الدالة في تغيير أى جزء داخل سلسلة حرفية معينة . والصورة العامة لها كالآتي :

STUFF(<exp1>,<start position>,<number of characters>;  
,<exp2>)

حيث

exp1 هي السلسلة الحرفية المطلوب التعديل فيها .

و exp2 هي السلسلة الحرفية المطلوب ادخالها في السلسلة الحرفية الأولى .

و start' position هي قيمة عددية تمثل المكان المطلوب ادخال السلسلة الحرفية بدلا منه .

number of characters هو عدد الحروف المطلوب استبدالها من السلسلة الحرفية الأولى . وإذا كان هذا العدد صفرا ، يتم ادخال السلسلة الحرفية الثانية داخل السلسلة الحرفية الأولى دون تغيير الحروف الموجودة في السلسلة الأولى . أى يتم حشر السلسلة الثانية داخل السلسلة الأولى (Insertion) . وإذا كانت السلسلة الثانية عبارة عن سلسلة خالية (null string) ، يتم مسح حروف من السلسلة الأولى بقدر طول السلسلة الخالية .

## مثال

إذا أريد تغيير عنوان معين داخل قاعدة بيانات الطلبة (Cadets) تكتب السطور التالية :

- . new\_street = nasr city
  - . USE Cadets
  - . GO 5
  - . ? address
- 10 -Ainshams -Cairo
- . REPLACE address WITH STUFF(address,4,9,new\_street)
  - . ?address

يلاحظ في هذه الحالة ظهور العنوان (10-nasr city-Cairo) بدلا من العنوان السابق .

## ٥٥ - الدالة (SUBSTR)

تستخدم هذه الدالة في استخراج جزء من سلسلة حرفية معينة . والصورة العامة لها كالآتي :

SUBSTR(<exp1>,<starting position>  
,[<number of characters>])

حيث (exp1) هو السلسلة الحرفية المطلوب استخراج جزء منها .

و (starting position) هو المكان الذي يبدأ منه استخراج السلسلة الحرفية الفرعية .

و (number of characters) هو عدد اختياري يمثل عدد الحروف المراد سحبها من السلسلة الحرفية . وفي حالة عدم كتابة هذا العدد يتم استخراج السلسلة الحرفية بدءا من مكان البداية (starting position) الى آخر السلسلة الحرفية .

## مثال

في المثال السابق الخاص بالعنوان الموجود في السجل الخامس . يراد معرفة المدينة التي يقع فيها هذا العنوان . لتنفيذ ذلك يتم كتابة السطر التالي :

?SUBSTR(address,14,5)

في هذه الحالة يظهر الآتي :

Cairo

## ٥٦ - الدالة (TIME)

هذه الدالة تعطى الوقت الحالي الذي تم ادخاله عند بدء تشغيل الجهاز من خلال نظام التشغيل . والصورة العامة لها كالآتي :

TIME()

مثال

للحصول على الوقت الحالي يتم كتابة السطر التالي :

?TIME()

وعند الضغط على مفتاح الادخال يظهر الآتي مثلا :

20:45:20

## ٥٧ - الدالة (TRANSFORM)

تستخدم هذه الدالة للتحكم في شكل المخرجات التي تنتج من الأوامر (???, DISPLAY, LABEL, LIST, REPORT) . والصورة العامة لها كالآتي :

TRANSFORM(<exp1> , <exp2>)

وهي تؤدي نفس العمل الذي تؤديه عبارة (PICTURE) مع الأمر (....SAY) .

مثال

لعرض الأسماء الموجودة في الحقل (name) بحيث يتم فصل كل حرف عن الحرف التالي بمسافة (Space) ، يتم كتابة السطر التالي :

DISPLAY TRANSFORM(name, @Rxxxxxxxxxx)

فإذا كان الاسم الموجود في هذا الحقل هو (Mohamed) مثلا يظهر الآتي :

M o h a m e d

ولعرض مرتب أربعة موظفين بحيث يتكون المرتب من ثمانية أرقام مع رقمين عشريين يتم كتابة السطر التالي :

LIST NEXT 4 TRANSFORM(salary, "# # # #.# # #")

في هذه الحالة تظهر الأعداد كالآتي :

Record #	Salary
1	570.50
2	600.80
3	1000.00
4	700.00

## ٥٨ - الدالة (TRIM)

تستخدم هذه الدالة في مسح المسافات الخالية من نهاية السلسلة الحرفية والصورة العامة لها كالآتي :

TRIM(<exp>)

وهي تفيد عندما يراد التأكد من التخزين الصحيح للبيانات التي يدخلها المستخدم ، حيث يتم أولا ادخال هذه البيانات في متغير ذاكرة بعد التخلص من المسافات الخالية في أوله أو في آخره ، ثم يتم نقل هذه البيانات من متغير الذاكرة الى الحقل الخاص بها .

فمثلا عندما يراد ادخال أى اسم في الحقل (name) ، يتم كتابة الأمر التالي لإنشاء متغير الذاكرة (mname)

STORE SPACE(30) TO mname

ثم يتم التخلص من المسافات الخالية في أول الاسم وآخره كالآتي :

```
STORE LTRIM(TRIM(mname)) TO mname
```

وهذا يؤدي إلى إدخال الحروف التي يدخلها المستخدم دون أي مسافات قبلها أو بعدها .

وتستخدم هذه الدالة أيضا عندما يراد معرفة الطول الصحيح لأي سلسلة حرفية باستخدام الدالة (LEN) وذلك كالآتي مثلا :

```
LEN(LTRIM(TRIM(mname)))
```

## ٥٩ - الدالة (TYPE)

تستخدم هذه الدالة في تحديد نوع أي قيمة أو متغير معين . والصورة العامة لها كالآتي :

```
TYPE(<exp>)
```

وهي تعطي حرف كبير (Capital) يمثل نوع القيمة (exp) ، مثل (C) للقيم الحرفية (Characters) ، (N) للقيم العددية (Numeric) ، (L) للقيم المنطقية (Logical) ، (M) لحقول الملاحظات (Memo) ، (U) للقيم غير المعرفة (Undefined) .

## مثال

عندما يراد اختبار المتغير (score) مثلا يتم كتابة السطر التالي :

```
? TYPE("score")
```

في هذه الحالة يظهر الحرف (U) ، وهذا يعني أن المتغير غير معرف (Undefined) . وذلك لأن المتغير لم يتم تعريفه قبل هذا الأمر . أما عند كتابة الآتي مثلا :

```
STORE 100 TO score
TYPE("score")
```

في هذه الحالة يظهر الحرف (N) ، وهذا يعني أن المتغير عددي .

## ٦٠ - الدالة (UPPER)

تستخدم هذه الدالة في تحويل الحروف الصغيرة (Lowercase) الى حروف كبيرة (Upercase) . والصورة العامة لها كالآتي :

UPPER(<exp>)

وتفيد هذه الدالة في التأكد من ادخال البيانات التي يدخلها المستخدم بنفس شكل البيانات المخزنة في الملف ، كما تفيد أيضا عندما يتم عرض رسالة على المستخدم واستقبال الرد على هذه الرسالة والتعامل مع هذا الرد بصرف النظر عن ادخاله بحروف كبيرة أو صغيرة .

## مثال

عندما يراد البحث عن اسم معين في حقل الاسم (name) ، يتم أولا انشاء متغير ذاكرة (Lookup) مثلا لاستقبال الاسم الذي يدخله المستخدم، ثم تحويل هذا المتغير الى حروف كبيرة حتى يماثل الحروف الموجودة في حقل الاسم لجميع السجلات ، ثم يتم البحث عن هذا الاسم باستخدام الأمر (SEEK) . ولتنفيذ ذلك يتم كتابة السطور التالية :

```
Lookup = SPACE(15)
@10,10 SAY "Enter name of person to edit" GET Lookup
READ
Lookup = UPPER(Lookup)
SEEK Lookup
```

## ٦١ - الدالة (VAL)

تستخدم هذه الدالة في تحويل الأرقام الموجودة في السلسلة الحرفية (String) الى العدد المقابل . والصورة العامة لها كالآتي :



VAL(<exp>)

وهي تقوم بعكس العمل الذي تؤديه الدالة (STR)، مع ملاحظة أن البيانات الموجودة في القيمة (exp) يجب أن تكون أعداداً وليست حروفاً .

## مثال

يمكن كتابة السطور التالية :

```
STORE "88.50" TO string
VAL(string)
```

في هذه الحالة يظهر نفس العدد (88.50) ولكن الفرق هنا أن هذه القيمة عددية ، أي يمكن التعامل معها بالجمع عليها أو الطرح منها، وهكذا .

## ٦٢ - الدالة (VERSION)

هذه الدالة تعطي رقم نسخة برنامج (DBaseIII+) أو أي برنامج آخر من برامج عائلة (DBase) مثل (DBaseIV) ، (FoxBase +) ، (FoxPro) المستخدمة . والصورة العامة لها كالآتي :

VERSION()

وتستخدم في البرامج التي تتطلب بعض الخصائص المرتبطة بنسخة البرنامج المستخدمة .

## ٦٣ - الدالة (YEAR)

تستخدم هذه الدالة في الحصول على العدد الممثل للسنة داخل تاريخ معين . والصورة العامة لها كالآتي :

YEAR(<exp>)

حيث (exp) هو متغير ذاكرة تاريخي أو حقل تاريخي أو تاريخ اليوم الحالي .

### مثال

إذا كان تاريخ اليوم الحالي هو (02/18/90) ، فيمكن الحصول على العدد الممثل للسنة كالآتي :

`?YEAR( DATE() )`

يلاحظ في هذه الحالة ظهور العدد (1990) .

ملحق ( ١ )

**أهم الأوامر والدوال المستخدمة**

**في برنامج (DBase IV)**



هذا الملحق يضم الأوامر والدوال الإضافية فقط ، حيث أن باقى الأوامر والدوال مطابقة تماما لما سبق عرضه في برنامج (DBase III +) .

## أولا : الأوامر (Commands)

### ١ - الأمر (???)

وهو يؤدي الى ارسال المخرجات مباشرة الى الطابعة .

### ٢ - الأمر (@... SAY... GET)

وهو نفس الأمر الخاص ببرنامج (DBase III+) مع اضافة امكانيات أخرى له مثل ادخال شروط (Conditions) لظهور بيانات ال (Gets) وظهور رسائل الأخطاء والتحكم في ألوان الأعمدة الضوئية . ولزيد من الايضاح أرجع الى الباب الخاص بالأوامر فى لكتاب الأول. والصورة العامة له كالآتى :

```
@ <ROW> , <COL>
[SAY <expression> [PICTURE <expC>]
[FUNCTION <function list>]]
[GET <variable>
[[OPEN WINDOW <window name>]
[PICTURE <expC>]
[FUNCTION <function list>]
[RANGE [<low>][,<high>]]
[VALID <condition> [ERROR <expC>]]
[WHEN <condition> [DEFAULT <expression>]
[MESSAGE <expC>]]
[COLOR [<stand>][,<enhanced>]]]
```

### ٣ - الأمر (@... FILL)

وهو يؤدي الى رسم مستطيل مظل بلون محدد

```
@ <row1> , <col1> FILL TO <row2> , <col2>
[COLOR <color attribute>]
```

## ٤ - الأمر (TO...@)

وهو نفس الأمر المستخدم في برنامج (DBase III+) لرسم مستطيل على الشاشة مع بعض الإضافات للتحكم في لون الخطوط. ارجع الى الباب الخاص بالأوامر في الكتاب الأول. والصورة العامة كالآتي :

```
@ <row1> , <col1> TO <row2> , <col2>
[ DOUBLE / PANEL / <border definition string>]
[ COLOR <color attribute> ]
```

## ٥ - الأمر (ACTIVATE MENU)

ويستخدم لتشغيل عمود قوائم (Menu Bar) سبق تصميمه .

## ٦ - الأمر (ACTIVATE POPUP)

ويستخدم لتشغيل عمود قوائم (Menu Bar) مع القوائم المتفرعة منه (POPUP).

## ٧ - الأمر (ACTIVATE SCREEN)

وهو يؤدي الى العودة الى الشاشة الكاملة بدلا من النافذة المفتوحة (Active Window).

## ٨ - الأمر (ACTIVATE WINDOW)

ويستخدم في تشغيل نافذة معينة بدلا من الشاشة الكاملة. والصورة العامة كالآتي :

```
ACTIVATE WINDOW <window name list> / ALL
```

## ٩ - الأمر (APPEND FROM ARRAY)

ويستخدم في اضافة سجلات الى ملف قاعدة البيانات من مصفوفة بيانات (Array). والصورة العامة كالآتي :

```
APPEND FROM ARRAY <array name>
[ FOR <condition> ]
```

#### ١٠ - الأمر (APPEND MEMO)

ويستخدم في نقل بيانات من ملف نصي (Text File) إلى حقل الملاحظات (Memo Field).

#### ١١ - الأمر (ASSIST)

وهو يؤدي إلى تشغيل مركز التحكم (Control Center) الخاص ببرنامج (DBase IV).

#### ١٢ - الأمر (BEGIN TRANSACTION)

ويستخدم في بدء تعديل بيانات ملف قاعدة البيانات المفتوح بناءً على بيانات ملفات الحركة . والصورة العامة كالآتي :

```
BEGIN TRANSACTION [<path name>]
<transaction commands>
END TRANSACTION
```

#### ١٣ - الأمر (CALCULATE)

ويستخدم في عمل حسابات مالية وإحصائية للبيانات العددية . والصورة العامة له كالآتي :

```
CALCULATE [ Scope ] <expression list>
[ FOR <condition> ]
[ WHILE <condition> ]
[ TO <memvar list> ]
[ TO ARRAY <array name> ]
```

حيث (expression list) هي أي عملية حسابية مثل (AVG) أي المتوسط أو (MAX) أي القيمة العظمى أو (MIN) أي القيمة الصغرى أو (STD) أي الانحراف المعياري أو ... الخ .

## ١٤ - الأمر (COMPILE)

وهو يستخدم في تحويل البرنامج المكتوب بشفرة المصدر (Source Code) الى برنامج منفذ (Executable Program) .

## ١٥ - الأمر (COPY TO ARRAY)

يستخدم في ملء مصفوفة معينة بالبيانات الموجودة في ملف قاعدة البيانات المفتوح . والصورة العامة له كالآتي :

```
COPY TO ARRAY <array name>
[ FIELDS <field list>]
[ FOR <condition>]
[ WHILE <condition>]
```

## ١٦ - الأمر (CREATE APPLICATION)

يستخدم لتشغيل مولد التطبيقات (Application Generator) الخاص ببرنامج (DBase IV) . وهو الذي يساعد على انشاء البرامج التطبيقية بواسطة برنامج (DBase IV) . والصورة العامة له كالآتي :

```
CREATE / MODIFY APPLICATION <file name> /?
```

## ١٧ - الأمر (DEACTIVATE MENU)

يستخدم في مسح عمود القوائم الموجود على الشاشة والاحتفاظ به في الذاكرة حتى يتم استخدامه مع الجمل الاختيارية (ON SELECTION)

## ١٨ - الأمر (DEACTIVATE POPUP)

يستخدم في مسح عمود القوائم والقوائم المتفرعة منه مع الاحتفاظ بها في الذاكرة .



## ١٩ - الأمر (DEACTIVATE WINDOW)

ويستخدم في مسح نوافذ معينة من الشاشة مع الاحتفاظ بها في الذاكرة .  
والصورة العامة له كالآتي :

```
DEACTIVATE WINDOW <window name list> /ALL
```

## ٢٠ - الأمر (DEBUG)

ويستخدم لتشغيل برنامج التصحيح (Debugger) الخاص ببرنامج (DBase IV) .  
والصورة العامة له كالآتي :

```
DEBUG <file name > / <procedure name>  
[ WITH <parameter list>]
```

## ٢١ - الأمر (DECLARE)

ويستخدم لإنشاء مصفوفة أحادية أو ثنائية الأبعاد .  
والصورة العامة له كالآتي :

```
DECLARE <array name1>  
[ <number of rows> ]  
[ <number of columns> ]  
<array name2>  
[ <number of rows> ]  
[ <number of columns> ] . . . . . ]
```

## ٢٢ - الأمر (DEFINE BAR)

ويستخدم هذا الأمر لتعريف أحد الاختيارات في عمود القوائم (Menu Bar)  
والصورة العامة له كالآتي :

```
DEFINE BAR <line number> of <popup name>  
PROMPT <expC>  
[ MESSAGE <expC> ]  
[ SKIP [ FOR <condition> ] ]
```

## ٢٣ - الأمر (DEFINE BOX)

ويستخدم هذا الأمر في إنشاء مستطيل حول البيانات والصورة العامة له كالآتي :

```
DEFINE BOX
    FROM <point column>
    TO <point column>
    HEIGHT <expN>
    [ AT LINE <point line> ]
    [SINGLE/DOUBLE/<border definition string>]
```

## ٢٤ - الأمر (DEFINE MENU)

ويستخدم هذا الأمر في إنشاء القوائم (Menus) . والصورة العامة له كالآتي :

```
DEFINE MENU <menu name>
    [MESSAGE <expC>]
```

## ٢٥ - الأمر (DEFINE PAD)

ويستخدم في تعريف أحد الاختيارات في القائمة الفرعية (Popup Menu) والصورة العامة له كالآتي :

```
DEFINE PAD <pad name> OF <menu name>
    PROMPT <expC>
    [AT <row> , <col>]
    [MESSAGE <expC>]
```

## ٢٦ - الأمر (DEFINE POPUP)

ويستخدم في تعريف النافذة الفرعية (Popup Window) . والصورة العامة له كالآتي :

```
DEFINE POPUP <popup name>
    FROM <row1> , <col1>
    [ TO <row2> , <col2> ]
    [PROMPT FIELD <field name>/PROMPT FILES]
    [ LIKE <skeleton> ]
    [PROMPT STRUCTURE ]
    [ MESSAGE <expC> ]
```

## ٢٧ - الأمر (DEFINE WINDOW)

ويستخدم في تعريف نافذة معينة متضمنا الحدود والألوان . والصورة العامة له كالآتي :

```
DEFINE WINDOW <window name>
      FROM <row1> , <col1>
      TO <row2> , <col2>
[ DOUBLE / PANEL / NONE / <border definition string> ]
[ COLOR [ <stand> ] [ , <enhanced> ] [ , <frame> ] ]
```

## ٢٨ - الأمر (LIST/DISPLAY FILES)

ويستخدم لعرض معلومات عن الفهرس الحالي . وهو مثل الأمر (DIR) في نظام التشغيل (DOS) .

## ٢٩ - الأمر (LIST/DISPLAY USERS)

ويستخدم لعرض حالة محطات العمل (Work Stations) التي تعمل مع البرنامج من خلال شبكة حاسبات (Network) .

## ٣٠ - الأمر (LOGOUT)

ويستخدم لفصل إحدى المحطات (Work Stations) المتصلة من خلال شبكة (Network) .

## ٣١ - الأمر (MOVE WINDOW)

ويستخدم لتحريك نافذة معينة على الشاشة . والصورة العامة له كالآتي :

```
MOVE WINDOW <window name>
      TO <row> , <column>
      / BY <delta row> , <delta column>
```

## ٢٢ - الأمر (ON PAD)

ويستخدم لربط قائمة فرعية معينة بعمود اختيارات معين . والصورة العامة له كالآتي :

```
ON PAD <pad name> OF <menu name>
[ ACTIVATE POPUP <popup name> ]
```

## ٢٣ - الأمر (ON PAGE)

ويستخدم للتحكم في ناقل الصفحة (Page Break) . وكذلك في العناوين العلوية (Headers) والعناوين السفلية (Footers) أثناء طباعة التقارير . والصورة العامة له كالآتي :

```
ON PAGE [ AT LINE <expC><command> ]
```

## ٢٤ - الأمر (ON READERROR)

ويستخدم في التعامل مع الأخطاء أثناء تشغيل البرنامج . والصورة العامة له كالآتي :

```
ON READERROR [ <command> ]
```

## ٢٥ - الأمر (ON SELECTION PAD)

ويستخدم في تنفيذ عمل معين عند اختيار أحد الاختيارات في إحدى القوائم الفرعية . والصورة العامة له كالآتي :

```
ON SELECTION PAD <pad name> OF <menu name>
[ <command> ]
```

## ٢٦ - الأمر (ON SELECTION POPUP)

ويستخدم في تنفيذ أمر معين عند اختيار أحد القوائم الفرعية (POPUP) . والصورة العامة له كالآتي :

```
ON SELECTION POPUP <popup name>
/ ALL [ <command> ]
```

## ٢٧ - الأمر (PLAY MACRO)

ويستخدم في تنفيذ مجموعة من الأوامر التي سبق تخزينها في مجموعة من المفاتيح (Macro) . والصورة العامة له كالآتي :

```
PLAY MACRO <macro name>
```

## ٢٨ - الأمر (PRINTJOB)

ويستخدم لتنفيذ بعض الأوامر أثناء تنفيذ عملية الطباعة . والصورة العامة له كالآتي :

```
PRINTJOB
    <commands>
ENDPRINTJOB
```

## ٢٩ - الأمر (PROTECT)

ويستخدم في تأمين بعض البيانات وحمايتها من الاستخدام الا بواسطة المختصين .

## ٤٠ - الأمر (RELEASE MODULE)

ويستخدم في مسح أى برنامج فرعى أو نافذة أو قائمة أو برنامج مكتوب بلغة التجميع من الذاكرة المؤقتة . والصورة العامة له كالآتي :

```
RELEASE MODULE [ <module name list> ]
    / MENUS [ <menu name list> ]
    / POPUPS [ <POPU name list> ]
    / WINDOWS [ <windows name list> ]
```

## ٤١ - الأمر (RESTORE MACRO)

ويستخدم لتحميل برنامج حروف مبرمجة (Macro) في الذاكرة المؤقتة . والصورة العامة له كالآتي :

```
RESTORE MACRO FROM <macro file>
```

## ٤٢ - الأمر (RESTORE WINDOW)

ويستخدم لتحميل نافذة من القرص الى الذاكرة المؤقتة . والصورة العامة له كالآتي :

```
RESTORE WINDOW <window name list>
/ ALL FROM <file name>
```

## ٤٣ - الأمر (ROLLBACK)

ويستخدم لاعادة ملف قاعدة البيانات الى حالته الأولى قبل ادخال حركة معينة (Transaction) . والصورة العامة له كالآتي :

```
ROLLBACK [<database filename>]
```

## ٤٤ - الأمر (SAVE MACROS)

ويستخدم لتخزين برنامج المفاتيح المبرمجة (Macro) في ملف معين . والصورة العامة له كالآتي :

```
SAVE MACROS TO <macro file>
```

## ٤٥ - الأمر (SAVE WINDOW)

ويستخدم لتخزين نافذة معينة في ملف .

## ٤٦ - الأمر (SCAN)

ويستخدم لتنفيذ بعض الأوامر على مجموعة من السجلات . والصورة العامة له كالآتي :

```
SCAN
[ <scope>]
[FOR <condition>]
[ WHILE <condition>]
[ <commands>]
[ LOOP]
[ EXIT]
ENDSCAN
```

#### ٤٧ - الأمر (SHOW MENU)

ويستخدم لعرض قائمة على الشاشة دون تشغيلها . والصورة العامة له كالآتي :

```
SHOW MENU <menu name>
[PAD <pad name>]
```

#### ٤٨ - الأمر (SHOW POPUP)

ويستخدم لعرض قائمة فرعية على الشاشة دون تشغيلها . والصورة العامة له كالآتي :

```
SHOW POPUP <popup menu>
```

#### ٤٩ - الأمر (UNLOCK)

ويستخدم لفتح الملف أو السجل حتى يصبح قابلا للتشغيل بواسطة المستخدم . والصورة العامة له كالآتي :

```
UNLOCK [ALL / IN <alias>]
```

## ثانيا : أوامر التجهيز (SET)

### ١ - الأمر (SET)

ويستخدم في عرض قوائم الضبط .

### ٢ - الأمر (SET AUTOSAVE)

ويستخدم في تخزين السجلات من المخزن المؤقت (Buffer) الى القرص .  
والصورة العامة له كالآتي :

SET AUTOSAVE ON/OFF

### ٣ - الأمر (SET BELL TO)

ويستخدم في التحكم في الصوت الذى يحدث عند تشغيل الجرس .  
والصورة العامة له كالآتي :

SET BELL TO [<frequency>,<duration>]

### ٤ - الأمر (SET BORDER TO)

ويستخدم في التحكم في حدود النوافذ والقوائم الفرعية .  
والصورة العامة له كالآتي :

SET BORDER TO  
[SINGLE/DOUBLE/PANEL/NONE  
/ <border definition string>]

### ٥ - الأمر (SET CLOCK)

ويستخدم في عرض ساعة النظام في الركن العلوى الأيمن من الشاشة .  
والصورة العامة له كالآتي :

SET CLOCK ON/OFF



## ٦ - الأمر (SET CLOCK TO)

ويستخدم في التحكم في مكان عرض ساعة النظام . والصورة العامة له كالآتي :

SET CLOCK TO [<row> , <column>]

## ٧ - الأمر (SET COLOR OF)

ويستخدم في تحديد ألوان أشياء محددة على الشاشة . والصورة العامة له كالآتي :

SET CLOCR OF NORMAL/MESSAGES  
/TITLES/HIGHLIGHT/BOX  
/INFORMATION/FIELDS  
TO [<attribute>]

## ٨ - الأمر (SET CURRENCY)

ويستخدم لعرض علامة العملة على اليمين أو اليسار حسب الحاجة . والصورة العامة له كالآتي :

SET CURRENCY LEFT/RIGHT

## ٩ - الأمر (SET DATE)

ويستخدم في تحديد طرق كتابة التاريخ طبقا للدول المختلفة . والصورة العامة له كالآتي :

SET DATE [TO] AMERICAN/ANSI/BRTISH  
/FRENCH/GERMAN/ITALIAN/JAPAN/USA  
/MDY/DMY/YMD

## ١٠ - الأمر (SET DEVELOPMENT ON/OFF)

ويستخدم عند تطوير البرنامج وعمل بعض التعديلات ، فعندما تكون شفرة المصدر (Source Code) قد تم تعديلها بتاريخ يلى تاريخ شفرة الهدف (Object Code) يقوم برنامج (DBase IV) بإعادة ترجمة البرنامج (Compiling) .

### ١١ - الأمر (SET DISPLAY TO)

ويستخدم لضبط البرنامج على حالة الشاشة (Display Mode) إذا كانت (Mono) أو (EGA) أو (VGA) . والصورة العامة له كالآتي :

SET DISPLAY TO MONO/COLOR/EGA 25  
/EGA 43/ MONO 43

### ١٢ - الأمر (SET ENCRYPTION)

ويستخدم في تشفير أو فك شفرة البرنامج الذي سبق حمايته بواسطة الأمر (PROTECT) . والصورة العامة له كالآتي :

SET ENCRYPTION ON/OFF

### ١٣ - الأمر (SET HOURS TO)

ويستخدم في عرض الساعة وتحديد عدد الساعات بها إذا كان (١٢) أو (٢٤) ساعة . والصورة العامة له كالآتي :

SET HOURS TO [12/24]

### ١٤ - الأمر (SET LOCK)

ويستخدم لتشغيل أو عدم تشغيل قفل السجل ، وهذا يعني التحكم في قدرة المستخدم على التعامل مع بيانات هذا السجل أو عدم قدرته على ذلك . والصورة العامة له كالآتي :

SET LOCK TO [<expC>]

### ١٥ - الأمر (SET MARK TO)

ويستخدم لتحديد الحرف الذي يفصل أرقام التاريخ ، والحرف المبدئي هو الحرف (/) .

## ١٦ - الأمر (SET PAUSE)

ويستخدم في إيقاف الشاشة أو عدم إيقافها عند عرض بيانات تزيد عن طول الشاشة . والصورة العامة له كالآتي :

SET PAUSE ON/OFF

## ١٧ - الأمر (SET POINT TO)

ويستخدم لتحديد الحرف الذي يمثل العلامة العشرية . والحرف المبدئي هو النقطة ( . ) . والصورة العامة له كالآتي :

SET POINT TO [<expC>]

## ١٨ - الأمر (SET PRECISION)

ويستخدم لتحديد عدد الكسور العشرية التي تظهر في الأعداد . والعدد المبدئي هو ( ١٦ ) والمدى من ( ١٠ ) إلى ( ٢٠ ) . والصورة العامة له كالآتي :

SET PRECISION TO [<expC>]

## ١٩ - الأمر (SET PRINTER)

ويستخدم في توجيه أو عدم توجيه المخرجات إلى الطابعة . والصورة العامة له كالآتي :

SET PRINTER ON/OFF

## ٢٠ - الأمر (SET PRINTER TO)

ويستخدم في توجيه المخرجات إلى طابعة محددة . والطابعة المبدئية هي ( PRN ) . والصورة العامة له كالآتي :

SET PRINTER TO <computer name>  
<printer name> = <destination>

**٢١ - الأمر (SET PRINTER TO FILE)**

ويستخدم لتوجيه المخرجات الى ملف بدلا من الطابعة . والصورة العامة له كالآتي :

SET PRINTER TO FILE <file name>

**٢٢ - الأمر (SET WINDOW OF MEMO)**

ويستخدم في تحديد النافذة التي يتم عن طريقها تعديل حقول الملاحظات (Memo Fields) . والصورة العامة له كالآتي :

SET WINDOW OF MEMO TO <window name>

## ثالثا : الدوال

### ١ - الدالة (ACCESS)

وهي دالة يتم عن طريقها تحديد مستويات التعامل مع البيانات للمستخدمين حسب درجة السرية الخاصة بكل منهم .

### ٢ - الدالة (ACOS)

وهي دالة تحسب مقدار الزاوية من جيب تمامها .

### ٣ - الدالة (ALIAS)

وهي دالة تحدد الاسم المرادف لمنطقة عمل محددة .

### ٤ - الدالة (ASIN)

وهي دالة تحسب مقدار الزاوية من جيبها .

### ٥ - الدالة (ATAN)

وهي دالة تحسب مقدار الزاوية من ظلها .

### ٦ - الدالة (ATN2)

وهي دالة تحسب مقدار الزاوية من جيبها وجيب تمامها . والصورة العامة له كالآتي :

$$ATN2 (<expN1> , <expN2>)$$

## ٧ - الدالة (BAR)

وهي دالة تعطي رقم آخر عمود قوائم (Menu Bar) ثم اختياره .

## ٨ - الدالة (CEILING)

وهي دالة تحدد أصغر قيمة تزيد عن أو تساوى القيمة المحددة في الدالة .  
والصورة العامة له كالآتي :

CEILING (<expN>)

## ٩ - الدالة (COMPLETED)

وهي دالة تحدد ما إذا كانت عملية ادخال الحركة (Transaction) قد تمت أم لا .

## ١٠ - الدالة (COS)

وهي دالة تحسب جيب تمام الزاوية .

## ١١ - الدالة (DIFFERENCE)

وهي تعطي الفرق بين سلسلتين حرفيتين . والصورة العامة لها كالآتي :

DIFFERENCE (<expC> , <expC>)

## ١٢ - الدالة (DTOR)

وهي دالة تقوم بتحويل الزاوية من الدرجات (Degrees) الى الوحدات القطرية (Radians) .

## ١٣ - الدالة (FILE)

وهي دالة تتحقق من وجود ملف معين .

#### ١٤ - الدالة (FKLABEL)

وهي تحدد اسم مفتاح وظيفة معين (Function Key) من رقمه . والصورة العامة له كالآتي :

FKLABEL (<expN>)

#### ١٥ - الدالة (FKMAX)

وهي دالة تحدد أكبر عدد من مفاتيح الوظائف يمكن برمجته .

#### ١٦ - الدالة (FLOCK)

وهي تستخدم في إغلاق ملف قاعدة البيانات حتى لا يستطيع أى شخص التعامل معه .

#### ١٧ - الدالة (FLOOR)

وهي دالة تحدد أكبر عدد أصغر من أو يساوى القيمة المحددة في الدالة .

#### ١٨ - الدالة (FV)

وهي دالة تعطى القيمة المستقبلية (Future Value) لاستثمار معين لفترة محددة وبمعدل ثابت . والصورة العامة له كالآتي :

FV (<payment> , <rate> , <period>)

#### ١٩ - الدالة (LASTKEY)

وهي دالة تعطى كود الآسكى (ASCII Code) لآخر مفتاح تم الضغط عليه للخروج من شاشة معينة .

## ٢٠ - الدالة (LIKE)

وهي دالة تقوم بمقارنة سلسلتين حرفيتين . والصورة العامة له كالآتي :

LIKE (<pattern> , <expC>)

## ٢١ - الدالة (LINNO)

وهي تحدد رقم السطر الذي سوف يتم تنفيذه من البرنامج .

## ٢٢ - الدالة (LOCK)

وتستخدم في إغلاق سجلات معينة من ملف قاعدة البيانات خلال شبكة الحاسبات (Network) حتى لا تستخدم بواسطة أى مستخدم . والصورة العامة له كالآتي :

LOCK ([<exp list>] [,<alias>])

## ٢٣ - الدالة (MENU)

وهي دالة تحدد اسم القائمة الفرعية المفتوحة .

## ٢٤ - الدالة (MESSAGE)

وهي دالة تحدد رسالة الخطأ الخاصة بآخر خطأ ظهر في البرنامج .

## ٢٥ - الدالة (MLINE)

وهي دالة تحدد سطرا معينا في حقل الملاحظات .

## ٢٦ - الدالة (NETWORK)

وهي دالة تحدد ما اذا كان البرنامج يعمل على الشبكة (Network) أم لا .



## ٢٧ - الدالة (PAD)

وهي تحدد اسم القائمة الجارى العمل عليها .

## ٢٨ - الدالة (PAYMENT)

وهي دالة تحدد الأقساط المدفوعة لسداد قرض معين فى فترة معينة وبفائدة معينة . والصورة العامة له كالاتى :

PAYMENT (<principal> , <rate> , <period>)

## ٢٩ - الدالة (PI)

وهي دالة تعطى النسبة التقريبية (ط) .

## ٣٠ - الدالة (POPUP)

وهي دالة تعطى اسم القائمة الفرعية المفتوحة .

## ٣١ - الدالة (PRINTSTATUS)

وهي دالة تعطى حالة الطابعة الموصلة .

## ٣٢ - الدالة (PROGRAM)

وهي دالة تعطى اسم البرنامج الجارى تنفيذه عندما يظهر خطأ معين . وذلك لتحديد البرنامج الفرعى المحتوى على الخطأ .

## ٣٣ - الدالة (PV)

وهي تعطى القيمة الحالية لقرض يتم تسديده على أقساط فى فترات زمنية محددة . والصورة العامة له كالاتى :

PV (<payment> , <rate> , <period>)

٢٤ - الدالة (RAND)

وهي دالة تقوم بتوليد أعداد عشوائية .

٢٥ - الدالة (RTOD)

وهي تستخدم لتحويل الزوايا من التقدير الدائري (Radian) الى درجات .

ملحق (٢)

**أهم الأوامر والدوال المستخدمة**

**في برنامج (FoxBase+)**



هذا الملحق يضم الأوامر والدوال الإضافية فقط . حيث ان باقي الاوامر و الدوال مطابقة تماما لما سبق عرضه في برامج (DBaseIII+) ، (DBaseIV) .

## أولا : الأوامر

### ١ - الأمر (CLEAR PROGRAM)

ويستخدم في مسح برنامج مكتوب بلغة التجميع (Assembly) .

### ٢ - الأمر (CLEAR TYPEAHEAD)

ويستخدم لمسح مخزن الذاكرة المؤقت (Buffer) .

### ٣ - الأمر (DELETE FILE)

ويستخدم في مسح ملف معين من القرص .

### ٤ - الأمر (DIMENSION)

ويستخدم في انشاء مصفوفة متغيرات أحادية أو ثنائية .

### ٥ - الأمر (DIR)

ويستخدم في عرض فهرس الملفات .

### ٦ - الأمر (FLUSH)

ويستخدم في نقل محتويات الذاكرة الى القرص .

### ٧ - الأمر (GATHER FROM)

ويستخدم في تخزين بيانات المصفوفة في متغيرات ذاكرة . والصورة العامة له

كالآتي :

GATHER FROM <array> [FIELDS <fields>]

#### ٨ - الأمر (MENU)

ويستخدم لتشغيل عمود اختيارات (Menu Bar) .

#### ٩ - الأمر (ON KEY = <expN>)

ويستخدم في تنفيذ مجموعة من الأوامر عند ضغط المستخدم على مفتاح محدد .

#### ١٠ - الأمر (RELEASE MODULE)

ويستخدم في مسح برنامج سبق تحميله في الذاكرة .

#### ١١ - الأمر (RESTORE SCREEN)

ويستخدم لتحميل شاشة من الذاكرة أو من متغيرات ذاكرة معينة .

#### ١٢ - الأمر (SAVE SCREEN)

ويستخدم لتخزين الشاشة في متغير ذاكرة محدد .

#### ١٣ - الأمر (SCATTER)

ويستخدم في نقل محتويات متغيرات ذاكرة إلى مصفوفة متغيرات . والصورة العامة له كالآتي :

SCATTER [FIELDS <fields list> ] TO <array>

#### ١٤ - الأمر (UNLOCK)

ويستخدم في مسح كل أقفال الملفات والسجلات حتى يمكن التعامل معها بواسطة أي مستخدم .

## ثانيا : الدوال

### ١ - الدالة (ALIAS)

وهي تعطى الاسم المرادف لمنطقة العمل المفتوحة .

### ٢ - الدالة (FCOUNT)

وهي تعطى عدد الحقول الموجودة في ملف قاعدة البيانات المفتوح .

### ٣ - الدالة (FILE)

وهي تختبر وجود ملف معين على القرص .

### ٤ - الدالة (FKLABEL)

وهي تعطى اسم مفتاح الوظيفة (Function Key) المقابل لرقم معين .

### ٥ - الدالة (FKMAX)

وهي تعطى أكبر عدد يمكن برمجته من مفاتيح الوظائف .

### ٦ - الدالة (FLOCK)

وهي تستخدم في اغلاق ملف قاعدة بيانات محدد حتى لا يتم استخدامه بواسطة باقي المستخدمين .

### ٧ - الدالة (LOCK)

وهي تستخدم في اغلاق السجل المفتوح حتى لا يتم استخدامه بواسطة المستخدمين الآخرين .

٨ - الدالة (SYS(0))

وهي تعطى اسم الجهاز ورقمه فى شبكة الحاسبات .

٩ - الدالة (SYS(1))

وهي تعطى التاريخ بالشكل المستخدم فى برنامج (FoxBase) وبرنامج (DBase II) .

١٠ - الدالة (SYS(2))

وهي تعطى عدد الثوانى المنقضية منذ منتصف الليل وحتى الوقت الحالى ، أى أنها تعطى الوقت الحالى .

١١ - الدالة (SYS(3))

وهي تعطى اسم ملف مؤقت يتم استخدامه بواسطة أحد المستخدمين المتصلين بالشبكة .

١٢ - الدالة (SYS(5))

وهي تعطى اسم وحدة الأقراص البدئية (Default) .

١٣ - الدالة (SYS(6))

وهي تعطى الطابعة الحالية (Current Printer) .

١٤ - الدالة (SYS(7) ، | W | )

وهي تعطى اسم ملف التشكيل المفتوح (Format File) فى منطقة العمل المثلة بالرقم (W) .



#### ١٥ - الدالة (SYS(9))

وهي تعطى رقم نسخة برنامج (FoxBase +) .

#### ١٦ - الدالة (SYS(10,d))

وهي تحول الرقم الممثل للتاريخ (d) الى تاريخ حرفي (Character Date) مطابق في الشكل للتاريخ المستخدم بواسطة برنامج (FoxBase) وبرنامج (DBase II) .

#### ١٧ - الدالة (SYS(11,s))

وهي عكس الدالة السابقة ، حيث تعطى الرقم الممثل للتاريخ الحرفي (s) .

#### ١٨ - الدالة (SYS(12))

وهي تعطى الذاكرة المؤقتة المتاحة .

#### ١٩ - الدالة (SYS(13))

وهي تعطى حالة الطابعة اذا كانت جاهزة أو غير جاهزة .

#### ٢٠ - الدالة (SYS(14,n,w))

وهي تعطى الملف الفهرسي رقم (n) في منطقة العمل (W) مع ملاحظة أن العدد (n) لايزيد عن (٧) والعدد (W) لايزيد عن (١٠) .

#### ٢١ - الدالة (SYS(15,t,s))

وهي دالة تعطى السلسلة الحرفية (s) المقابلة للسلسلة الحرفية (t) حسب جدول التحويل الخاص المستخدم بالترجمة (Translation Table) .

#### ٢٢ - الدالة (SYS(16,n))

وهي دالة تعطى اسم البرنامج الجارى تنفيذه ، وهي تفيد عند متابعة خطأ

معين ومحاولة التوصل الى البرنامج المتسبب في هذا الخطأ . والعدد (n) يمثل المستوى الذى يقع عليه البرنامج ، فالعدد ( ١ ) يمثل البرنامج الرئيسى (Master Program) ، وإذا تم حذف العدد (n) فان الدالة تعطى البرنامج الحالى الجارى تنفيذه . وإذا أريد متابعة جميع البرامج الفرعية يتم استخدام الأوامر التالية :

```
STORE 1 TO i
DO WHILE LEN(SYS(16,i)<>0)
    ? SYS(16,i)
    STORE i+1 TO i
ENDDO
```

### ٢٣ - الدالة (SYS(17))

وهي تعطى نوع المعالج الدقيق المستخدم فى الجهاز اذا كان (8086/8088) أو (80286) أو (80386) .

### ٢٤ - الدالة (SYS(18))

وهي تعطى اسم حقل قاعدة البيانات الذى يتم ادخاله عن طريق الأمر (@...GET) ، وذلك عند ضغط المستخدم على المفتاح الذى تم تحديده بواسطة الأمر (ON KEY=t) حيث (t) يمثل اسم المفتاح .

### ٢٥ - الدالة (SYS(100))

وهي تعطى حالة الشاشة (Console) اذا كانت (ON) أو (OFF) .

### ٢٦ - الدالة (SYS(101))

وهي تعطى حالة الجهاز المستخدم اذا كان الشاشة أو الطابعة .

### ٢٧ - الدالة (SYS(102))

وهي تعطى حالة الطابعة (Printer) اذا كانت (ON) أو (OFF) .

٢٨ - الدالة (SYS(103))

وهي تعطى حالة الأمر (SET TALK) إذا كانت (ON) أو (OFF) .

٢٩ - الدالة (UPDATED)

وهي تعطى حالة البيانات إذا كان قد تم تحديثها أم لا .

٣٠ - الدالة (VERSION)

وهي تعطى رقم نسخة برنامج (FoxBase +) المستخدم .



ملحق (٢)

**أهم الأوامر و الدوال المستخدمة**

**في برنامج (FoxPro)**



هذا الملحق يضم الأوامر والدوال الإضافية فقط ، حيث أن باقي الأوامر والدوال مطابقة تماما لما سبق عرضه في برامج (FoxBase +) ، (DBase IV) ، (DBase III +) .

## أولا : الأوامر

### ١ - الأمر (COPY MEMO)

يستخدم لنسخ محتويات حقل الملاحظات في ملف .

### ٢ - الأمر (FILER)

يستخدم في تشغيل برنامج صيانة الملفات الخاص ببرنامج (FoxPro) .

### ٣ - الأمر (FOR)

يستخدم في انشاء حلقة تكرارية مع تشغيل عداد لحساب عدد مرات تنفيذ الحلقة . والصورة العامة له كالآتي :

```
FOR <memory>=<expN1>TO<expN2>[STEP<expN3>]<statements>
    [EXIT]
    [LOOP]
ENDFOR
```

### ٤ - الأمر (FUNCTION)

وهو أمر يحدد بداية برنامج فرعي (Subroutine) . والصورة العامة له كالآتي :

```
FUNCTION <name>
```

### ٥ - الأمر (MODIFY MEMO)

يستخدم في تعديل محتويات حقل الملاحظات .

- ٦ - الأمر (SET COLOR SET)   
 ويستخدم في تحميل مجموعة من الألوان سبق تحديدها .
- ٧ - الأمر (SET CURSOR)   
 ويستخدم في التحكم في مؤشر الشاشة .
- ٨ - الأمر (SET LOGERRORS)   
 ويستخدم في ارسال أخطاء البرنامج أثناء ترجمته الى ملف على القرص .
- ٩ - الأمر (SET MACKEY TO)   
 ويستخدم في تحديد مفتاح أو عدة مفاتيح لتشغيل برنامج الماكرو .
- ١٠ - الأمر (SET MOUSE)   
 ويستخدم في تشغيل الفأرة (Mouse) .
- ١١ - الأمر (SET MOUSE TO)   
 ويستخدم في التحكم في الفأرة (Mouse) .
- ١٢ - الأمر (SET RELATION OFF INTO)   
 ويستخدم في الفاء علاقة بين ملفين مفتوحين .
- ١٣ - الأمر (SET SHADOWS)   
 ويستخدم في التحكم في ظهور ظل النافذة أو عم ظهوره .



## ثانيا : الدوال

### ١ - الدالة (ALLTRIM)

وتستخدم في حذف المسافات الخالية من أول السلسلة الحرفية وآخرها .

### ٢ - الدالة (ATC)

وتستخدم في البحث عن سلسلة حرفية داخل سلسلة حرفية أخرى بصرف النظر عن حالة الحروف إذا كانت صغيرة أو كبيرة ، وهي تعطي عددا يمثل مكان هذه السلسلة . والصورة العامة له كالآتي :

ATC (<expC1> , <expC2>[ , <expN>])

### ٣ - الدالة (ATCLINE)

وهي تبحث عن سلسلة حرفية معينة داخل سلسلة حرفية أخرى بصرف النظر عن حالة الحروف إذا كانت صغيرة أو كبيرة ، وهي تعطي رقم السطر المحتوي على هذه السلسلة . والصورة العامة له كالآتي :

ATCLINE (<expC1> , <expC2>)

### ٤ - الدالة (ATN2)

وهي تعطي قيمة الزاوية بالتقدير الدائري من جيب الزاوية وجيب تمامها . والصورة العامة له كالآتي :

ATN2 (<expN1> , <expN2>)

### ٥ - الدالة (CAPSLOCK)

وهي تعطي حالة مفتاح الحروف الكبيرة (Caps) .

#### ٦ - الدالة (CHRSAW)

وهي تحدد ما اذا كان أحد الحروف موجودا في مخزن الذاكرة الخاص بلوحة المفاتيح .

#### ٧ - الدالة (CHRTRAN)

وهي دالة تقوم بترجمة حرف أو حروف معينة باستخدام جداول ترجمة محددة (Translation Tables) . والصورة العامة له كالآتي :

CHRTRAN (<expC1> , <expC2> , <expC3>)

#### ٨ - الدالة (CURDIR)

وهي تعطي الفهرس الحالي (Current Directory) .

#### ٩ - الدالة (DATE)

وهي تعطي تاريخ اليوم الحالي .

#### ١٠ - الدالة (DMY)

وهي دالة تقوم بتحويل التاريخ الى الشكل (Day Month Year) .

#### ١١ - الدالة (EMPTY)

وهي تحدد اذا كانت السلسلة خالية (Blank) أم لا .

#### ١٢ - الدالة (FCOUNT)

وهي تعطي عدد الحقول في ملف قاعدة البيانات المفتوح .

١٢ - الدالة (FGETS)

وهي تعطى حروفا من ملف حتى يتم الضغط على مفتاح الإدخال .

١٤ - الدالة (FILTER)

وهي تعطى الشروط المستخدمة في المرشح (Filter) المستخدم .

١٥ - الدالة (FOPEN)

وتستخدم في فتح ملف .

١٦ - الدالة (FPUTS)

وهي تقوم بكتابة سلسلة حرفية في الملف المفتوح .

١٧ - الدالة (FREAD)

وهي تقوم بقراءة عدد من الحروف من الملف المفتوح .

١٨ - الدالة (FSIZE)

وهي تعطى حجم حقل معين بالحروف (Bytes) .

١٩ - الدالة (FULLPATH)

وهي تعطى المسار الخاص بملف معين .

٢٠ - الدالة (FWRITE)

وهي تكتب سلسلة حرفية معينة في الملف المفتوح .

٢١ - الدالة (HEADER)

وهي تعطى عدد الحروف الموجودة في العنوان (Header) الخاص بأحد الملفات .

٢٢ - الدالة (INLIST)

وهي تحدد ما إذا كانت سلسلة حرفيه معينة موجودة ضمن مجموعة من السلاسل الحرفية .

٢٣ - الدالة (ISDIGIT)

وهي تحدد ما إذا كان أول حرف من سلسلة حرفية رقما أم حرفا .

٢٤ - الدالة (MCOL)

وهي تعطى مكان العمود الخاص بمؤشر نافذة معينة .

٢٥ - الدالة (MDY)

وهي تحول التاريخ الى صورة ( شهر يوم سنة ) أى (Month, Day, Year) .

٢٥ - الدالة (MEMLINES)

وهي تعطى عدد السطور في حقل الملاحظات .

٢ - الدالة (MEMORY)

وهي تعطى الذاكرة الموقتة المتاحة .

٢٧ - الدالة (MESSAGE)

وهي تعطى رسالة الخطأ المستخدمة .

## ٢٨ - الدالة (MROW)

وهي تعطي رقم العمود الخاص بمؤشر نافذة معينة .

## ٢٩ - الدالة (OCCURS)

وهي تعطي عدد مرات وجود سلسلة حرفية معينة داخل سلسلة حرفية أخرى .  
والصورة العامة له كالآتي :

OCCURS (<expC1> , <expC2>)

## ٣٠ - الدالة (ORDER)

وهي تعطي اسم الفهرس الرئيسى فى منطقة عمل محددة .

## ٣١ - الدالة (PARAMETERS)

وهي تعطي عدد المعاملات التى تم ادخالها الى آخر برنامج خطوات  
(Procedure).

## ٣٢ - الدالة (PROPER)

وهي تعطي الأسماء بشكلها المتعارف عليه . وهو أن يكون أول حرف كبيرا  
(Capital) وباقي الحروف صغيرة (Small) .

## ٣٣ - الدالة (RATLINE)

وهي تبحث عن وجود سلسلة حرفية معينة داخل سلسلة حرفية أخرى وتحدد  
رقم السطر الموجودة به . والصورة العامة له كالآتي :

RATLINE (<expC1> , <expC3>)

#### ٣٤ - الدالة (RELATION)

وهي تعطى العلاقة التي سبق تكوينها في منطقة العمل المفتوحة .

#### ٣٥ - الدالة (RTOD)

وهي تحول الزاوية من التقدير الدائري الى درجات .

#### ٣٦ - الدالة (SCOLS)

وهي تعطى عدد الأعمدة المتاحة على الشاشة .

#### ٣٧ - الدالة (SIGN)

وهي تحدد الإشارة الخاصة بالعدد سواء كانت سالبة أو موجبة .

#### ٣٨ - الدالة (BROWS)

وهي تعطى عدد السطور المتاحة على الشاشة .

#### ٣٩ - الدالة (SYS(23))

وهي تعطى حجم الذاكرة الممتدة (Extended) المستخدمة في برنامج (FoxPro) .

#### ٤٠ - الدالة (SYS(24))

وهي تعطى حجم الذاكرة الممتدة (Extended) المتاح والذي سبق تحديده من خلال ملف المواصفات (CONFIG.SYS) .

#### ٤١ - الدالة (SYS(1001))

وهي تعطى حجم الذاكرة المتاحة لبرنامج (FoxPro) .

- ٤٢ - الدالة (SYS(2000,<expC>))  
وهي تعطي اسم الملف المطابق للسلسلة الحرفية (<expC>).
- ٤٣ - الدالة (SYS(2001,<expC>))  
وهي تعطي الحالة الحالية (Current Status) للأمر (SET) المثل بالسلسلة الحرفية (<expC>).
- ٤٤ - الدالة (SYS(2002))  
وهي تتحكم في تشغيل أو عدم تشغيل مؤشر الشاشة (Cursor).
- ٤٥ - الدالة (SYS(2003))  
وهي تعطي اسم الفهرس الحالي المستخدم.
- ٤٦ - الدالة (SYS(2005))  
وهي تعطي اسم ملف الموارد المستخدم (Resource File).
- ٤٧ - الدالة [SYS(2006)]  
وهي تعطي نوع الشاشة المستخدمة والكارت المستخدم معها.
- ٤٨ - الدالة (SYS(2007,<expC>))  
وهي تعطي نتيجة اختبار الجمع (Check Sum) للسلسلة الحرفية <expC>.
- ٤٩ - الدالة (SYS(2008))  
وهي تحدد شكل مؤشر التصحيح المستخدم.

**٥٠ - الدالة (SYS(2009))**

وهي تحول بين وضع الاضافة (Insert) ووضع الكتابة الفوقية (Overwrite).

**٥١ - الدالة (WCOLS)**

وهي تعطى عدد الأعمدة في نافذة معينة .

**٥٢ - الدالة (WEXIST)**

وهي تحدد ما اذا كانت نافذة معينة قد تم تعريفها أم لا .

**٥٣ - الدالة (WLCOL)**

وهي تعطى رقم العمود الخاص بنافذة معينة .

**٥٤ - الدالة (WLROW)**

وهي تعطى رقم الصف الخاص بنافذة معينة .

**٥٥ - الدالة (WOUTPUT)**

وهي تحدد ما اذا كانت المخرجات قد تم توجيهها الى نافذة معينة أم لا .

**٥٦ - الدالة (WROWS)**

وهي تعطى عدد السطور في نافذة معينة .





## موسوعة "دلتا كمبيوتر" لتكنولوجيا وعلوم الحاسب

تعتبر المكتبة العربية ومحتوياتها في مجال التكنولوجيا من أكبر الدعامات الأساسية لشعرة والتي تشكل بدورها أحد العوامل الرئيسية لجوانب التنمية المختلفة في المنطقة العربية . ولما كانت تكنولوجيا الحاسبات من أهم اتجاهات المعرفة التكنولوجية في الآونة الأخيرة فإن قيمة المؤلفات تزدهر في هذا الجانب من واقع ازدياد حاجة المستخدم العربي إليها . وبما لا شك فيه أن المكتبة العربية في مجال تكنولوجيا وعلوم الحاسب تعتبر ذخيرة في هذا النوع من المؤلفات إلى درجة بعيدة نظراً لعدد جرائب تذكر منها مايلي :

- العمق الفنى اللازم والمواكبة للتطور التكنولوجى السريع .
- افتقار المكتبة العربية الى التقدر المطلوب من البعد العلمى اللازم للبعد الفنى .
- الترابط الكامل بين جوانب المعرفة فى المراجع المختلفة وعلاقات ذلك بدرجة استفادة القارئ والتعكاس على درجة المعرفة ومستوى الخبرة .
- درجة ارتباطها بالتطبيق ومستوى استفادة القارئ منها .
- التغطية الكاملة لكل مستويات القراء مع اختلاف ثقافتهم وخبراتهم .
- حاجة القارئ العربى فى هذه المرحلة تتجاوز مستوى العديد من المراجع المتاحة والتي تعتمد على الترجمة الحرفية لدليل التشغيل للنظم التكنولوجية المختلفة الخاصة بالحاسب .
- ومن ههنا المنطلق فقد قامت مؤسسة دلتا باعداد موسوعة " دلتا كمبيوتر " لتكنولوجيا وعلوم الحاسب - والتي تتكون من العديد من المراجع - على ايدى نخبة مختارة من أساتذة الجامعات وكبار الخبراء المتخصصين فى هذا المجال .
- ومع التطور السريع فى عالم تكنولوجيا الحاسبات وتعدد جوانب المعرفة المطلوبة للقارئ العربى فإن موسوعة دلتا قد تم اعدادها على اساس التغطية الشاملة لاتجاهات التكنولوجيا الحديثة تبعاً لتأثيرات المطروحة مع التغطية المستمرة للمستجدات فى هذا المجال من خلال الاصدارات المختلفة لكاتب الموسوعة على ضوء التطور السريع فى مجال تكنولوجيا الحاسبات .

١ - الحاسبات الالكترونية حاضرها ومستقبلها

٢ - الموسوعة الشاملة لمصطلحات الحاسب الالكترونى

٣ - المرجع الشامل لنظام التشغيل (DOS)

MS DOS  
MS WINDOWS NORTON UTILITIES  
PC TOOLS VIRUS-SCAN

٤ - عالم الجداول الالكترونية

LOTUS 123  
EXCEL  
QUATRO PRO

٥ - نظم ادارة قواعد البيانات ( الجزء الاول )

FOXBASE+ DBASE III+  
FOXPRO DBASE IV

٦ - نظم ادارة قواعد البيانات ( الجزء الثانى )

٧ - تطبيقات نظم ادارة قواعد البيانات

٨ - فيروسات الحاسب وأمن البيانات

٩ - النظم الحاسبية والحاسب الالكترونى

حسابات المخازن - الحسابات العامة

حسابات العملاء - حسابات المورد

حسابات الربح

١٠ - الحاسب الالكترونى وإدارة المشروعات

١١ - النظم الخفيفة والكثيفة الاصطناعى



دلتا كمبيوتر  
Delta Computer



